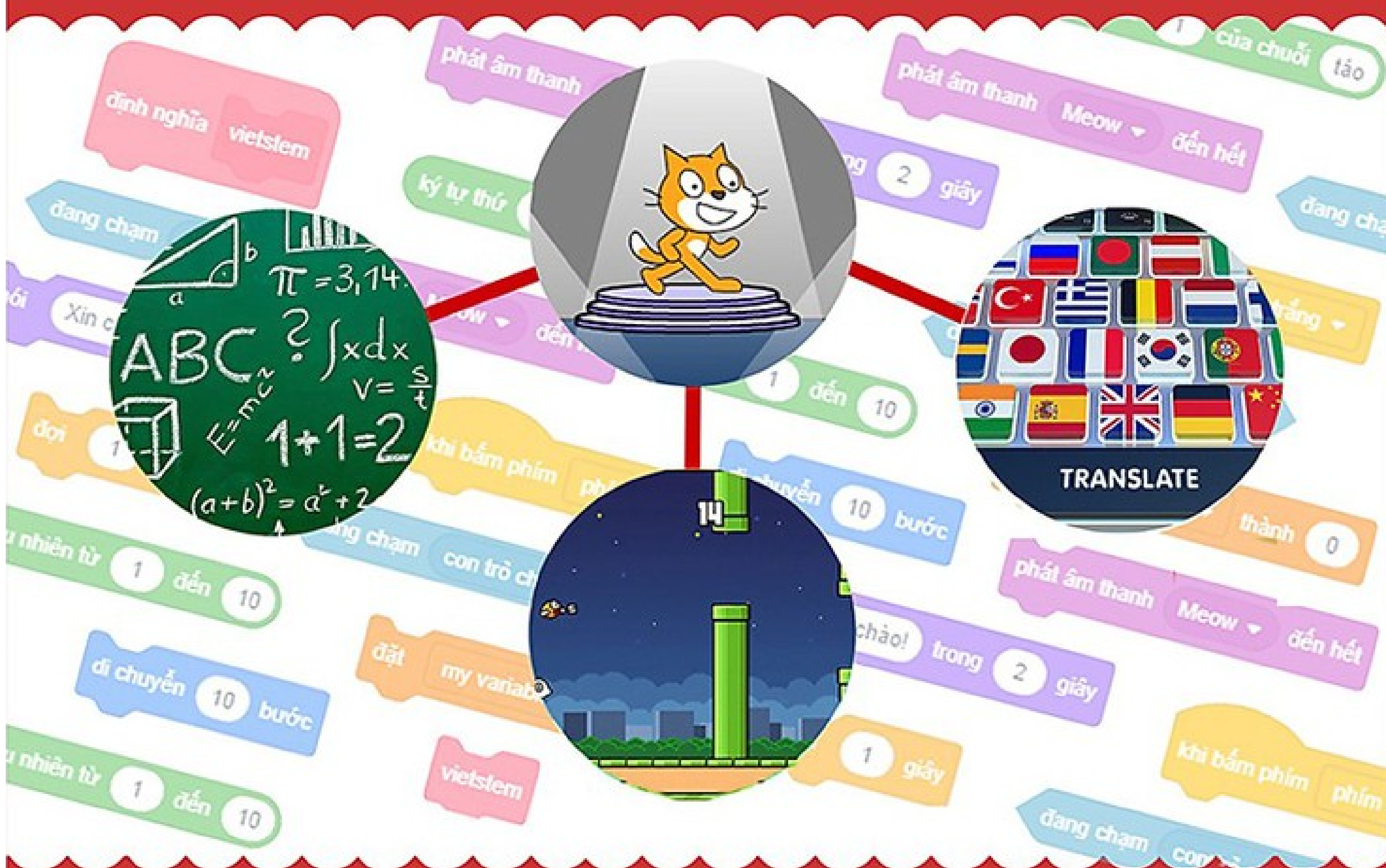
NGUYỄN HỮU HƯNG (Chủ biên)
TẠ BÍCH - DƯƠNG LỰC - NGUYỄN MINH - NGUYỄN HOÀN

LẬP TRÌNH với SCRATCH 3.0

HÀNH TRANG CHO TƯƠNG LAI



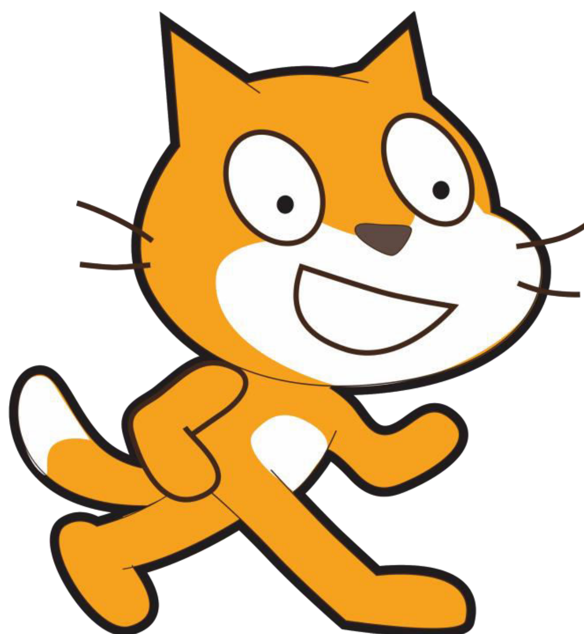
LẬP TRÌNH

với SCRATCH 3.0

CHƯƠNG I

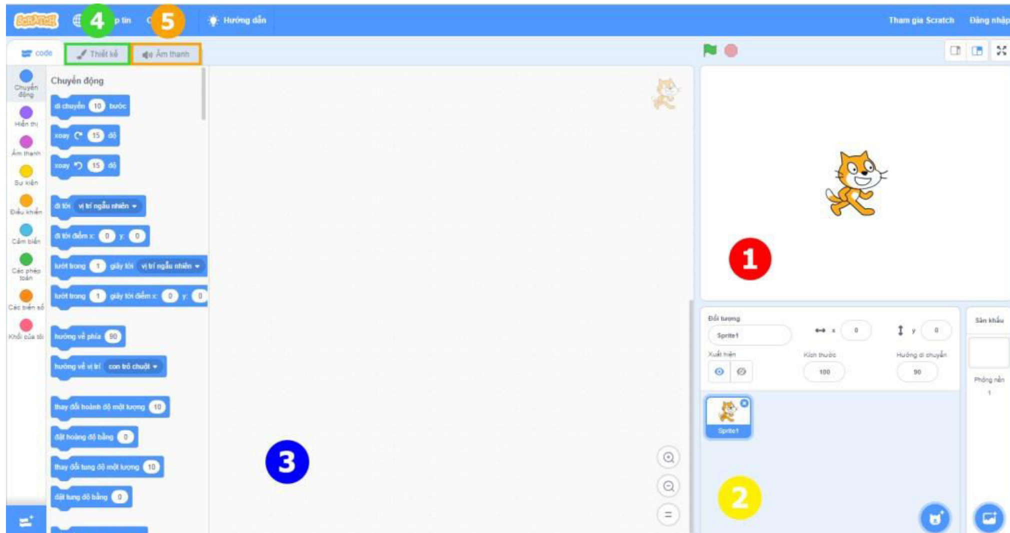
Phần mềm Scratch 3.0

Trong chương này, các bạn sẽ được tìm hiểu về phần mềm Scratch 3.0, các thành phần của giao diện và các chức năng của từng phần. Sau khi kết thúc chương, các bạn sẽ có được cái nhìn tổng quan và cách sử dụng một số chức năng cơ bản của phần mềm.



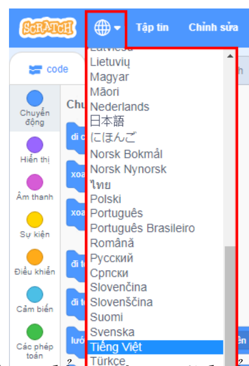
BÀI 1 GIAO DIỆN

Giao diện mặc định của Scratch 3.0.



Hình 1.1.1. Giao diện phần mềm Scratch 3.0

Ngôn ngữ hiển thị ngầm định của chương trình là tiếng Anh. Các bạn có thể chuyển đổi giao diện sang bản Việt hoá bằng cách nháy chuột vào biểu tượng quả địa cầu, di chuyển chuột xuống dưới và chọn Tiếng Việt, khi đó ngôn ngữ hiển thị trên giao diện hầu hết sẽ là tiếng Việt.

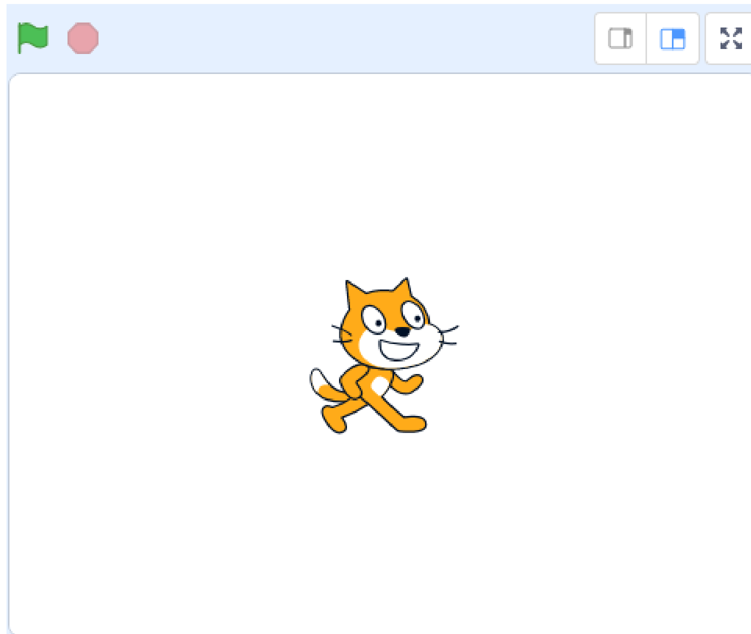


Hình 1.1.2. Thay đổi ngôn ngữ hiển thị trên giao diện

Giao diện Scratch 3.0 có năm khu vực chính như mô tả trên *hình 1.1.1.*

1.1. Khu vực sân khấu

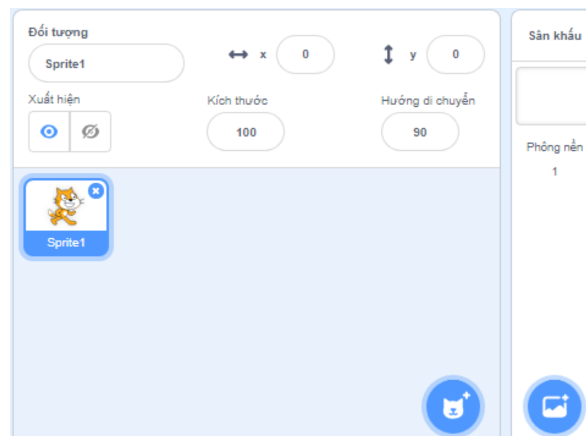
Khu vực sân khấu (được đánh số 1) là nơi biểu diễn từng phần hoặc cả dự án. Khi khởi động phần mềm, một chú mèo xuất hiện trên sân khấu.



Hình 1.1.3. Khu vực sân khấu

1.2. Khu vực quản lí phông nền sân khấu và nhân vật

Khu vực này gồm các biểu tượng thu nhỏ như khung chứa phông nền sân khấu và khung nhân vật (gồm danh sách các nhân vật hoặc đối tượng trong chương trình).



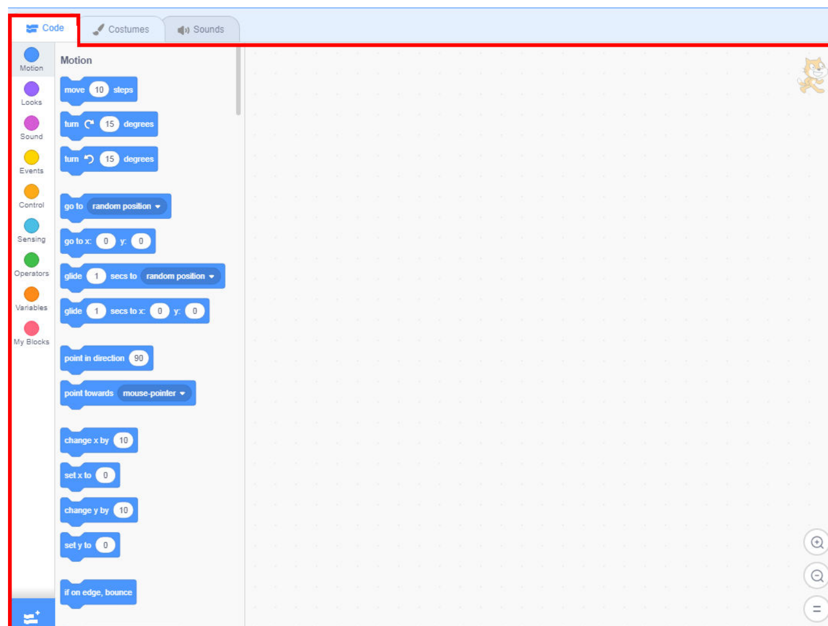
Hình 1.1.4. Khu vực quản lí phông nền sân khấu và nhân vật

1.2.1. Nhân vật

Trong khung nhân vật có thể có nhiều nhân vật. Có ba khu tác động vào nhân vật, đó là Khu lập trình ❸, Khu ngoại hình ❹ và Khu âm thanh ❺.

a) Khu lập trình

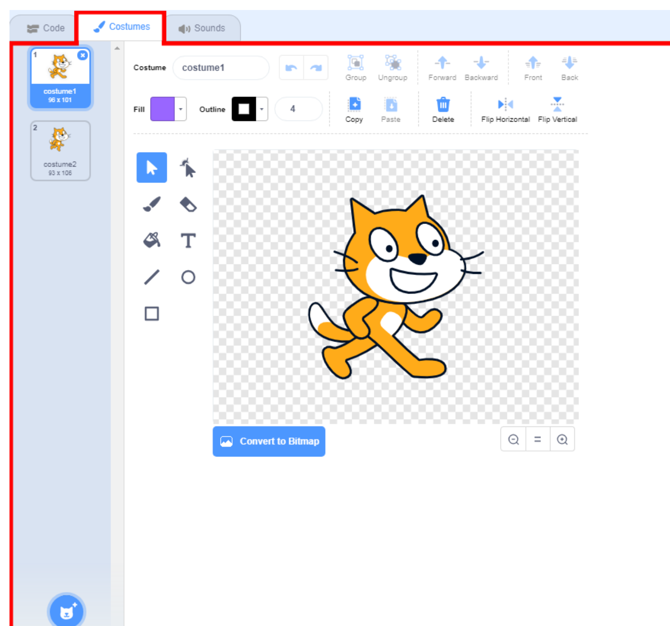
Hiện ra khi chọn trang **Code**, là nơi các khối lệnh được ghép với nhau tạo thành một chương trình theo ý muốn của lập trình viên.



Hình 1.1.5. Khu lập trình

b) Khu ngoại hình

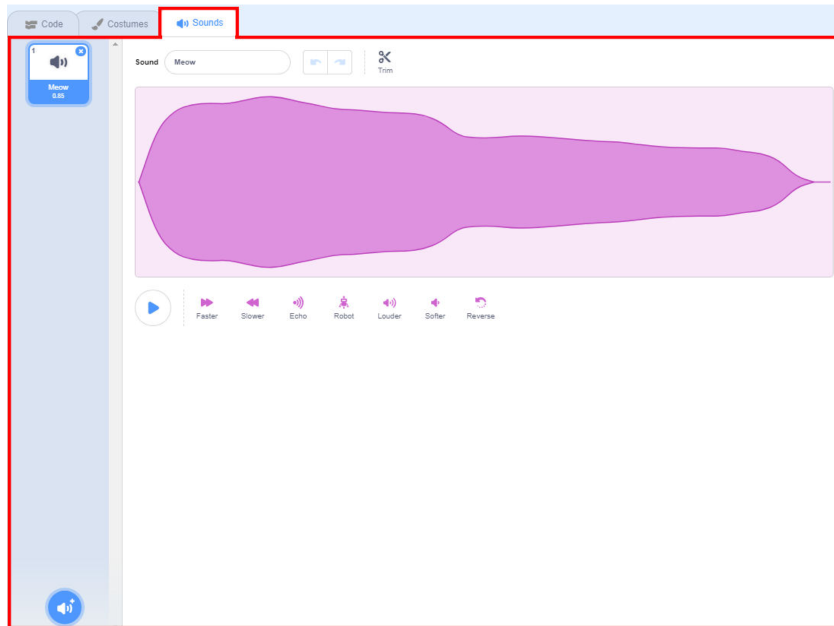
Mỗi nhân vật sẽ có một hoặc nhiều hình ảnh hiển thị. Khu ngoại hình sẽ xuất hiện khi chọn trang **Thiết kế**, cho phép các lập trình viên thao tác với hình ảnh của nhân vật như thêm mới, chỉnh sửa, xoá các hình ảnh trên cơ sở thư viện có sẵn hoặc tự tạo theo yêu cầu.



Hình 1.1.6. Khu ngoại hình

c) Khu âm thanh

Xuất hiện khi chọn trang **Âm thanh**, cho phép lập trình viên tạo, chỉnh sửa âm thanh cho các nhân vật trên cơ sở thư viện có sẵn hoặc tự tạo theo yêu cầu.



Hình 1.1.7. Khu âm thanh

1.2.2. Phông nền sân khấu

Phông nền sân khấu có thể coi là một đối tượng (nhân vật) đặc b//iệt, cũng có ba khu tác động vào, đó là Khu lập trình, Khu phông nền và Khu âm thanh. Về cơ bản, ba khu này có thành phần và chức năng tương đương với ba khu tác động vào nhân vật, chỉ có một chút khác biệt:

a) Khu lập trình

Giống với Khu lập trình của nhân vật, tuy nhiên vì sân khấu không thể di chuyển nên số lượng các khối lệnh sẽ ít hơn.

b) Khu phông nền

Khu phông nền của sân khấu xuất hiện khi chọn trang **Thiết kế**, là nơi quản lí các phông nền (Khu này cũng chính là Khu ngoại hình của nhân vật).

c) Khu âm thanh

Giống hoàn toàn với khu âm thanh của nhân vật.

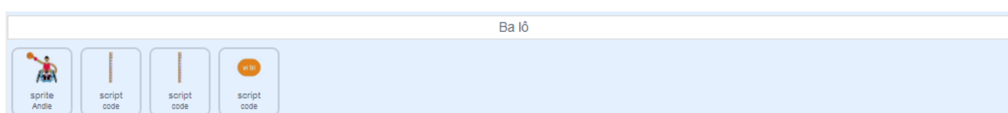
1.3. Thanh danh mục, Khu lưu trữ và Thư viện hướng dẫn

Ngoài năm khu vực chính, giao diện phần mềm Scratch còn có Thanh danh mục, và Thư viện video hướng dẫn sẽ được nhắc đến trong những bài sau.



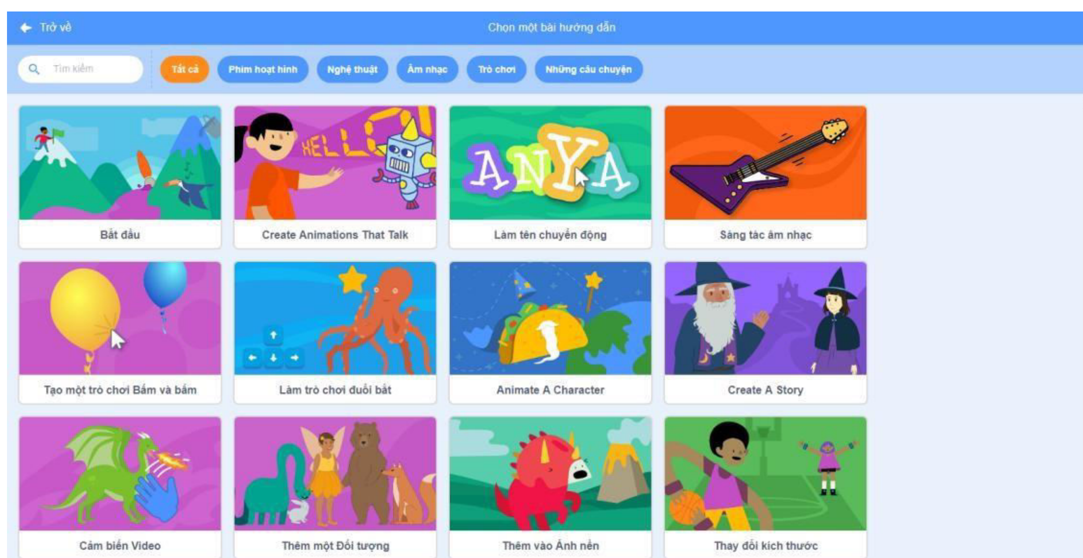
Thanh danh mục

Hình 1.1.8. Thanh danh mục



Hình 1.1.9. Khu lưu trữ





Hình 1.1.10. Thư viện video hướng dẫn

BÀI 2 THANH DANH MỤC VÀ KHU LƯU TRỮ

2.1. Thanh danhmục

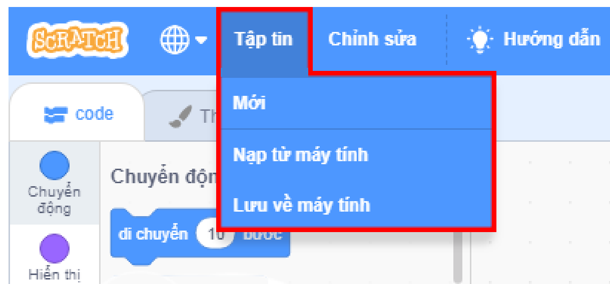


Hình 1.2.1. Thanh danh mục

Thanh danh mục nằm ở phía trên cùng của giao diện và có vai trò rất quan trọng liên quan tới thao tác sử dụng chương trình Scratch. Sau đây là từng thành phần của Thanh danh mục:

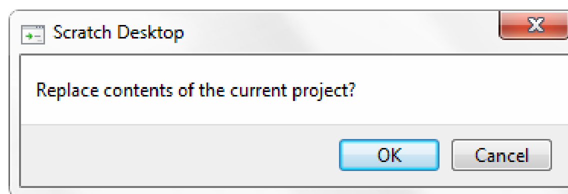
1. Tập tin

Đây là nơi thao tác với tập tin dự án. Tập tin là một tập hợp các thông tin được lưulại trên bộ nhớ máy tính. Ví dụ một bức ảnh, một bài hát, một bộ phim,... chúng ta đều gọi đó là tập tin. Mỗi dự án Scratch cũng là một tập tin.



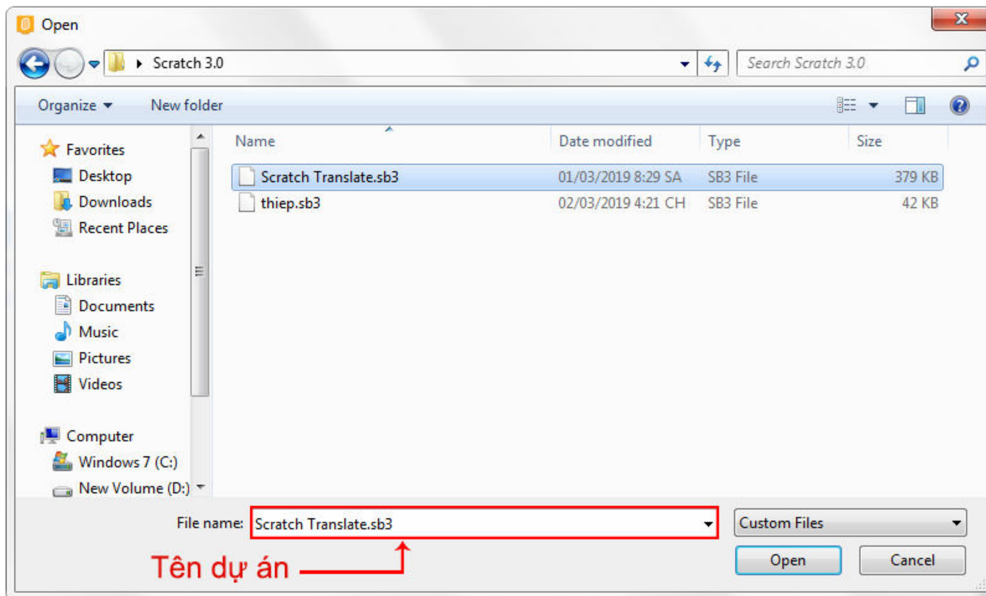
Hình 1.2.2. Tập tin

- Lựa chọn **Mới**: Lựa chọn này sẽ tạo một dự án mới để làm việc. Khi các bạn đang làm việc với một dự án khác mà chưa lưu lại, nếu chọn **Mới** thì sẽ có mộtcửa sổ hiện ra hỏi bạn có muốn thay thế nội dung của dự án hiện tại hay không. Chọn OK dự án mới xuất hiện và dự án cũ sẽ không được lưulại.



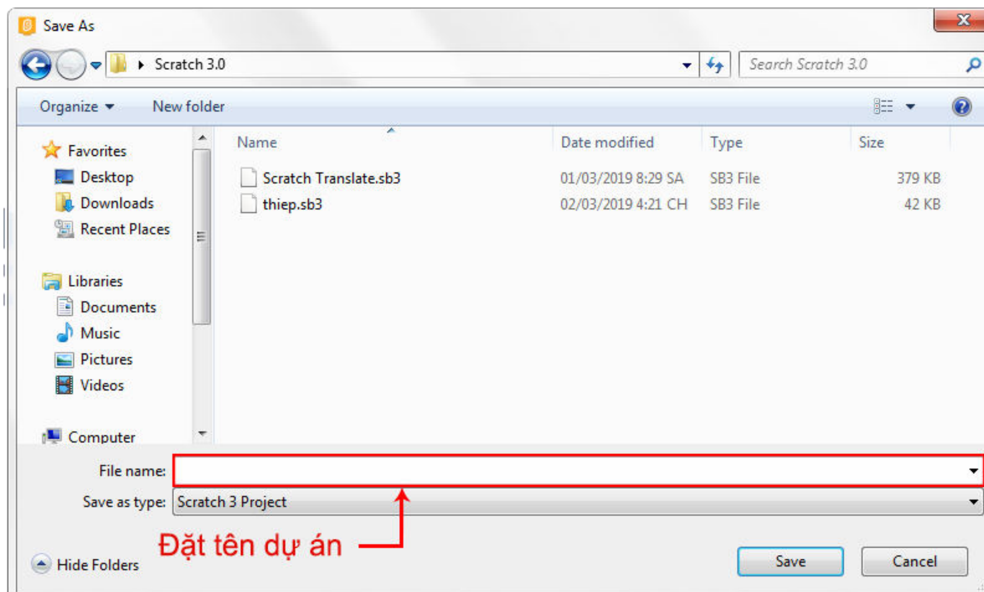
Hình 1.2.3. Cửa sổ xác nhận thay thế nội dung dự án hiện tại

- Lựa chọn **Nạp từ máy tính**: Dùng để mở một dự án đã có sẵn trên máy tính, một cửa sổ sẽ hiện ra để các bạn chọn tập tin dự án trên máy tính.



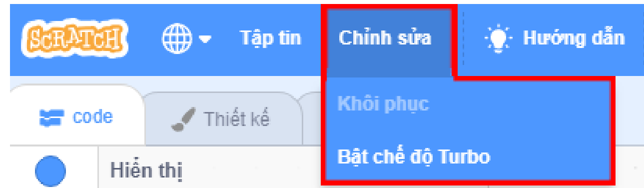
Hình 1.2.4. Cửa sổ mở dự án

- Nếu chọn **Lưu về máy tính** thì một cửa sổ khác sẽ hiện ra để các bạn đặt tên và lựa chọn vị trí lưu lại trên máy tính.



Hình 1.2.5. Cửa sổ lưu dự án

2. Chỉnh sửa

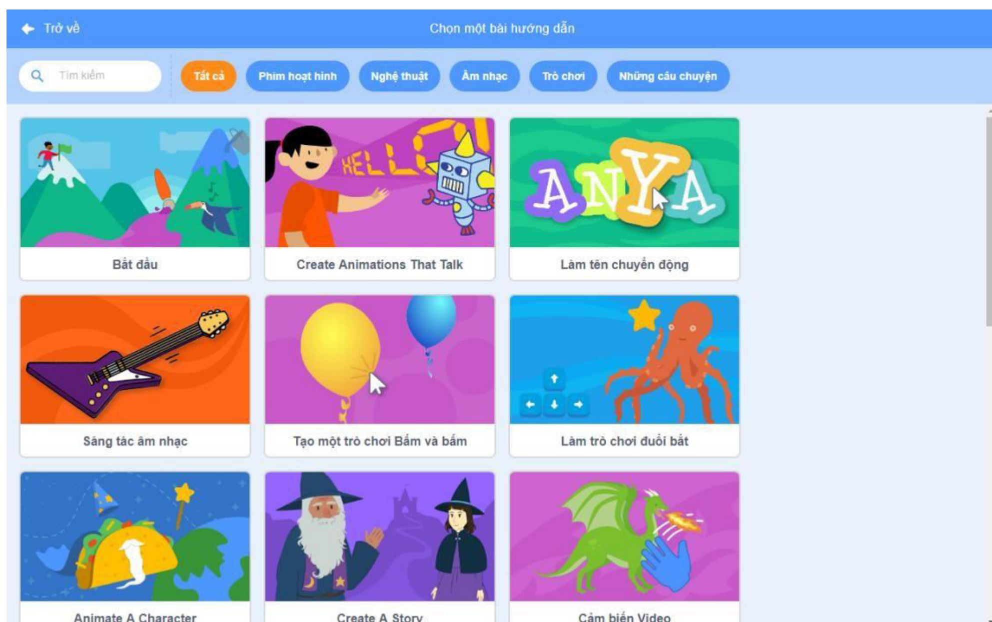


Hình 1.2.6. Chỉnh sửa

- Lựa chọn **Khôi phục**: Lấy lại đối tượng hoặc khối lệnh vừa bị xoá đi. Lựa chọn này chỉ lấy lại được đối tượng hoặc khối lệnh bị xoá gần nhất.
- Lựa chọn **Turbo mode**: Bật/Tắt chế độ Turbo. Chế độ này giúp chương trình tăng tốc độ xử lí các khối lệnh.

3. Hướng dẫn

Khi chọn mục này, thư viện hướng dẫn sẽ hiện ra, chứa những dự án mẫu được thể hiện qua các video hướng dẫn.



Hình 1.2.7. Thư viện hướng dẫn

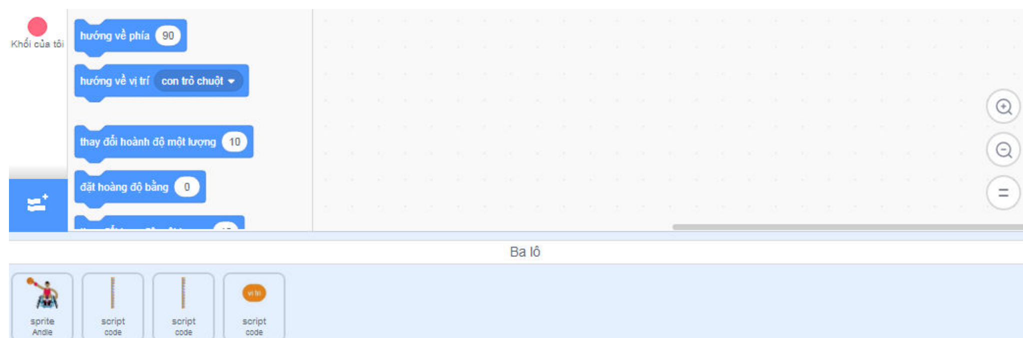
2.2. Khu lưu trữ

Khu lưu trữ (Ba lô) chỉ xuất hiện trong Scratch 3.0 online, có vị trí nằm ở phía dưới của khu lập trình. Khu lưu trữ là nơi lưu lại những nhân vật khối lệnh hoặc đoạn khối lệnh đã được tạo trước đó.



Hình 1.2.8. Khu lưu trữ

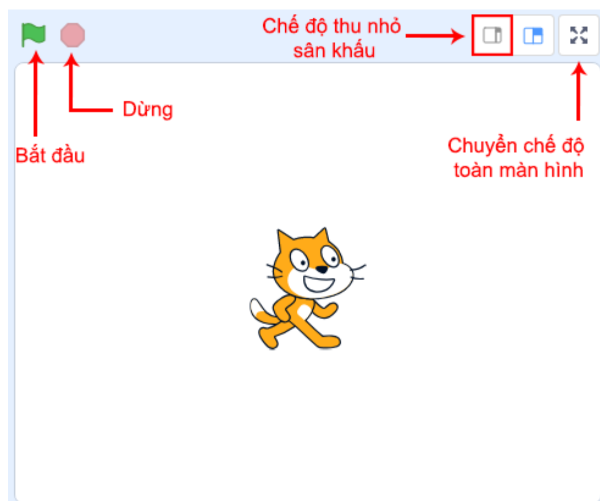
Khu lưu trữ là nơi lưu lại những đối tượng, khối lệnh hoặc đoạn khối lệnh được tạo trước đó.



Hình 1.2.9. Đối tượng, khối lệnh, đoạn khối lệnh được lưu lại

BÀI 3 SÂN KHẤU

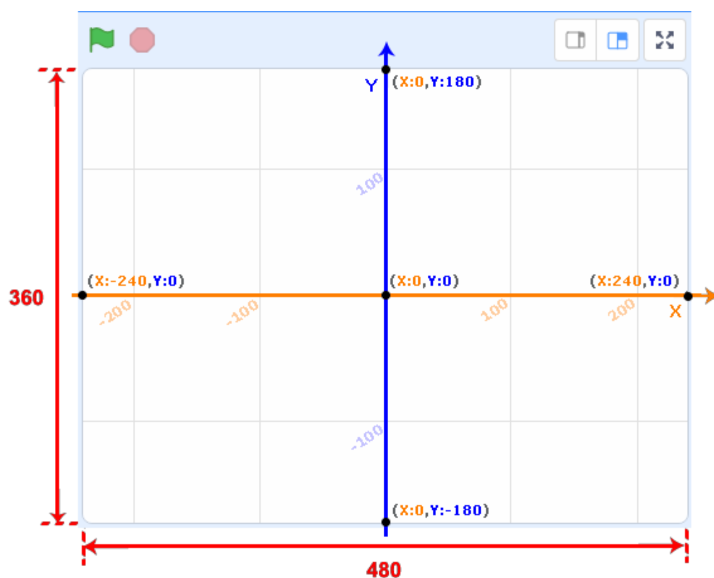
Sân khấu là nơi biểu diễn các hành động, hình ảnh, âm thanh của các đối tượng khi chạy chương trình, là nơi hiển thị kết quả của dự án.



Hình 1.3.1. Khu vực sân khấu

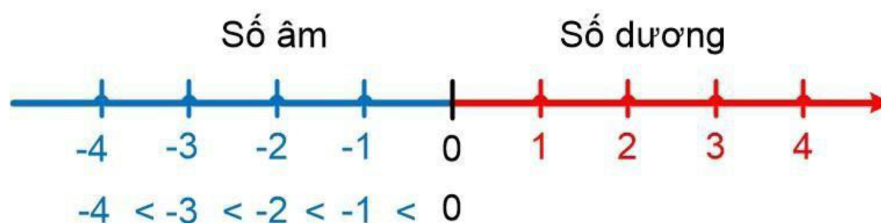
3.1. Hệ toạ độ trong sân khấu

Sân khấu có chiều rộng là 360 và chiều dài là 480, được xác định theo hệ trục toạ độ *Oxy* trong toán học. Trục *x* theo chiều ngang có giá trị từ –240 đến 240, trục *y* theo chiều dọc có giá trị từ –180 đến 180.



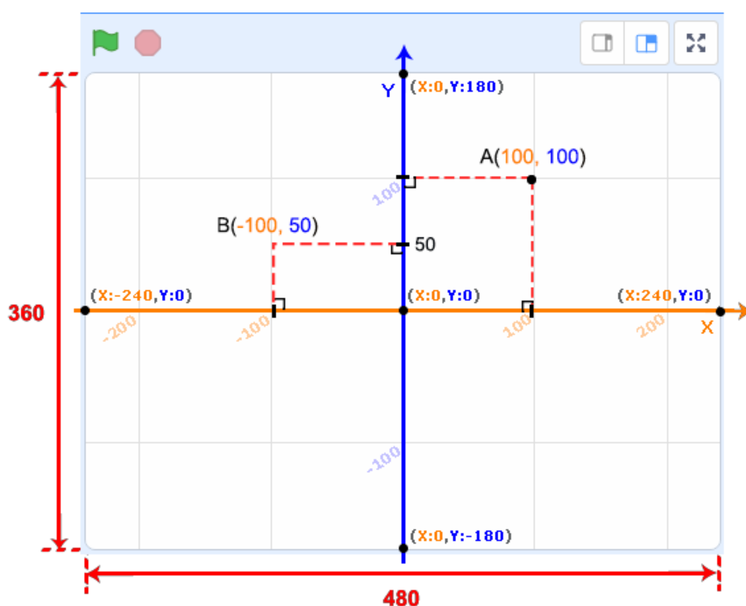
Hình 1.3.2. Hệ toạ độ trong sân khấu

Trong hệ toạ độ có sử dụng đến số âm. Số âm là số nhỏ hơn 0, được biểu diễn bằng dấu trừ (–) đứng trước số dương tương ứng. Trên trục số thì số âm đứng bên trái số 0.



Hình 1.3.3. Trục số

Toạ độ của một điểm được xác định bởi toạ độ theo trục *x* và toạ độ theo trục *y* (gọi tắt là toạ độ *x* và toạ độ *y*) của điểm đó. Để xác định toạ độ của một điểm trên sân khấu, từ điểm đó ta kẻ một đường thẳng vuông góc với trục *x*, đường thẳng đó cắt trục *x* tại điểm có *x* bằng bao nhiêu thì đó sẽ là vị trí *x* của điểm đó. Tương tự như với trục *y*, ta tìm được vị trí *y*. Cặp số *(x; y)* chính là toạ độ của điểm đó. Trục *x* và trục *y* giao nhau tại điểm có toạ độ (0; 0).



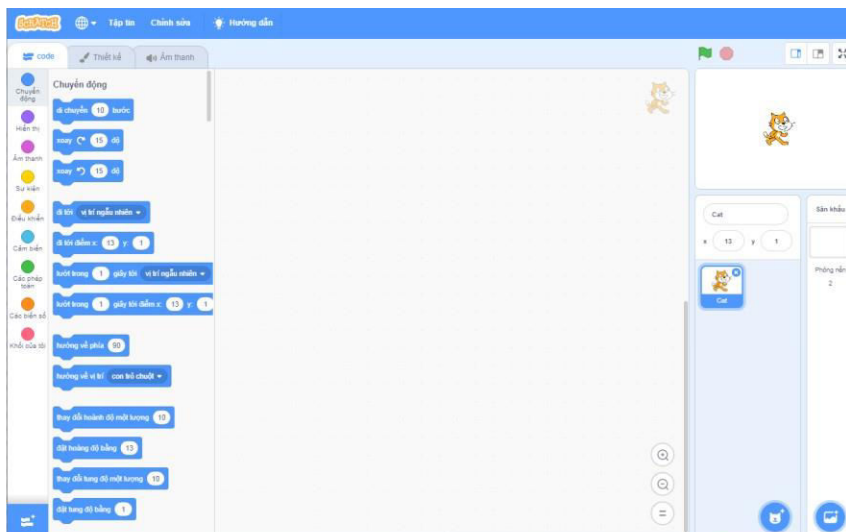
Hình 1.3.4. Xác định toạ độ của một điểm

3.2. Các chế độ của sân khấu

Sân khấu có ba chế độ: **chế độ ngầm định**, **chế độ thu nhỏ** và **chế độ toàn màn hình**.

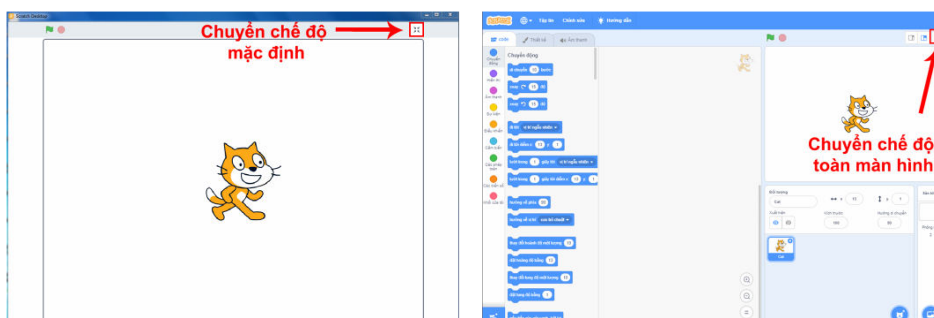
1. Chế độ ngầm định: là chế độ thiết đặt sẵn khi mở chương trình.

2. Chế độ thu nhỏ: Ở chế độ này, chiều dài và chiều rộng của sân khấu đều nhỏ bằng một nửa so với chế độ ngầm định. Để chuyển sân khấu về chế độ này, các bạn vào mục **Chỉnh sửa** trên **Thanh danh mục**, chọn **Small stage layout**. Chọn lại một lần nữa để sân khấu chuyển về chế độ ngầm định như ban đầu.



Hình 1.3.5. Chế độ thu nhỏ

3. Chế độ toàn màn hình: Khu vực sân khấu được phóng to toàn màn hình. Để chuyển sân khấu về chế độ này, các bạn nháy chuột vào nút ở góc trên bên phải sân khấu.



Hình 1.3.6. Chế độ mặc định và chế độ toàn màn hình của sân khấu

Dù ở chế độ nào thì hệ toạ độ trên sân khấu vẫn không thay đổi, trục x từ –240 đến 240, trục y từ –180 đến 180.

Ở góc trên bên trái sân khấu có hai nút và . Khi bạn chọn nút lệnh thì các khối lệnh được ghép với khối lệnh Khi bấm vào trong nhóm **Sự kiện** (Khu lập trình) sẽ được thực hiện. Biểu tượng này thường dùng để kích hoạt khối lệnh bắt đầu dự án.

Khi chọn nút lệnh thì tất cả hoạt động trên sân khấu sẽ dừng lại. Có thể ngừng chương trình vào thời điểm bất kì.

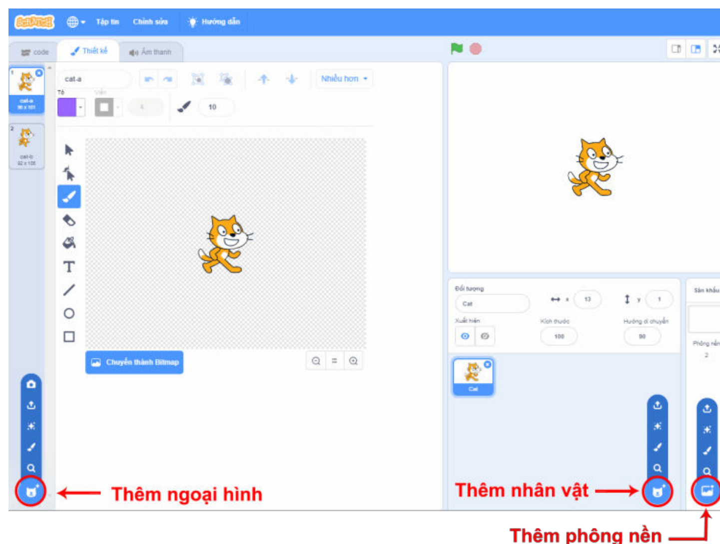
Khi chương trình đang chạy thì biểu tượng **Bắt đầu** có dạng , trong khi biểu tượng **Dừng** có dạng (đậm hơn). Khi chương trình dừng thì biểu tượng Bắt đầu không còn nền xanh đậm bao quanh và biểu tượng **Dừng** mờ đi.



Hình 1.3.7. Biểu tượng Bắt đầu và Dừng

BÀI 4 LÀM VIỆC VỚI PHÔNG NỀN, NHÂN VẬT VÀ NGOẠI HÌNH NHÂN VẬT

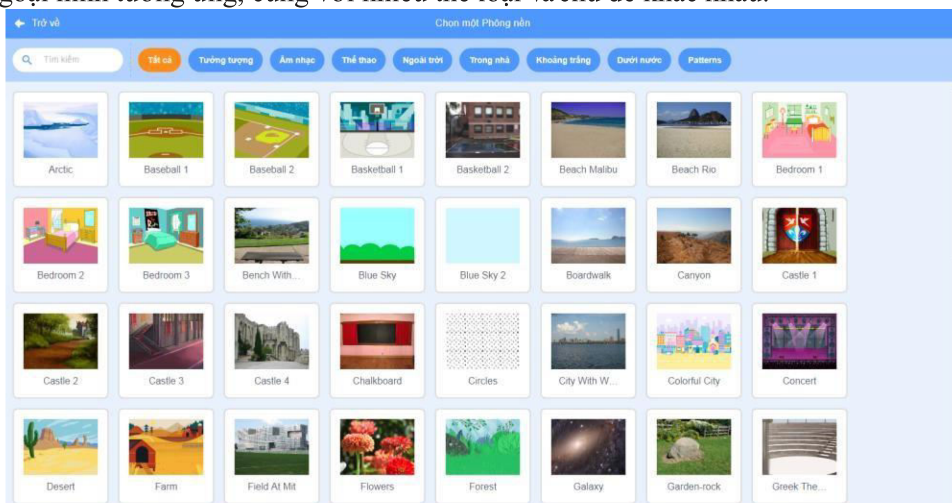
Việc thêm phông nền, nhân vật và ngoại hình nhân vật đều có 4 cách chính:



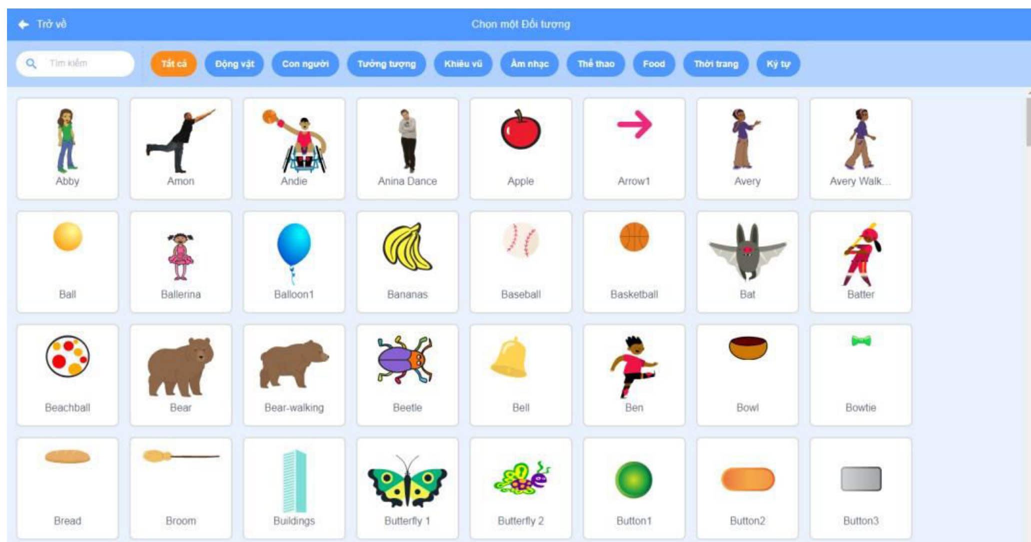
Hình 1.4.1. Thêm phông nền, nhân vật và ngoại hình nhân vật

4.1. Chọn đối tượng từ thư viện

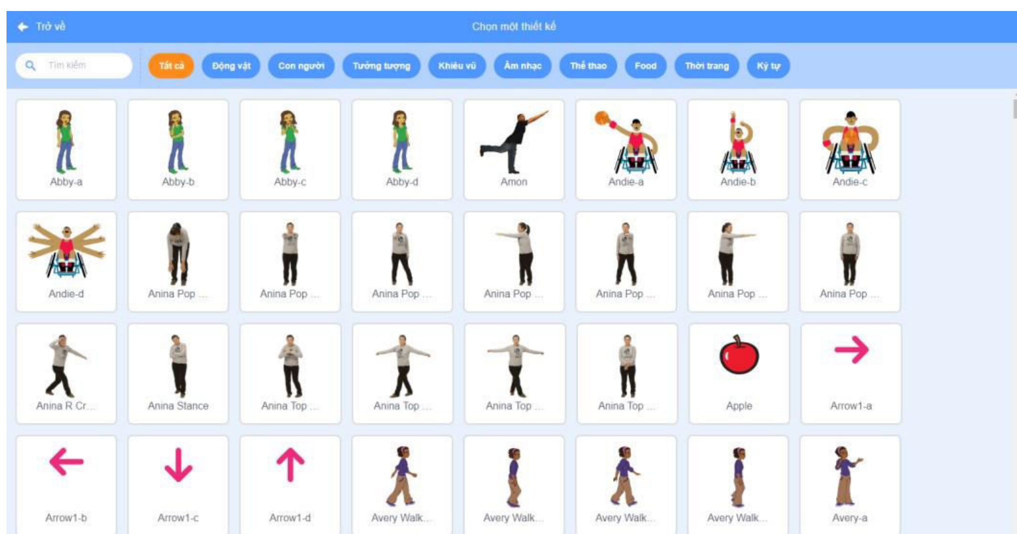
Biểu tượng của lệnh cho phép chọn đối tượng từ thư viện khi muốn thêm phông nền, nhân vật hay ngoại hình là .Bạn nháy chuột vào biểu tượng đó sẽ mở ra thư viện của phông nền hay nhân vật và ngoại hình tương ứng, cùng với nhiều thể loại và chủ đề khác nhau.



Hình 1.4.2. Thư viện phông nền



Hình 1.4.3. Thư viện nhân vật



Hình 1.4.4. Thư viện ngoại hình

4.2. Vẽmới

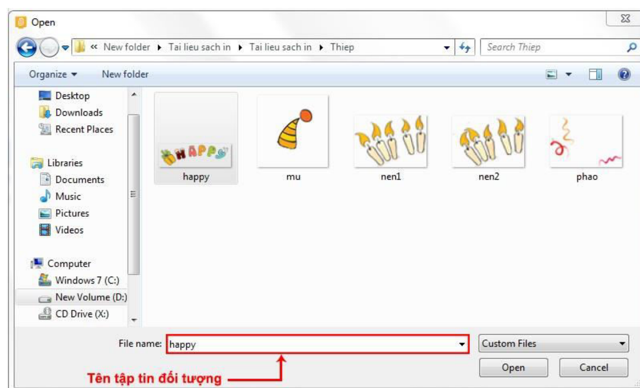
Khi chọn lệnh để vẽ mới , Khu thiết kế xuất hiện cùng với các công cụ để bạn tự vẽ, tự thiết kế đối tượng mới cho riêng mình. Chúng ta sẽ làm quen với Khu thiết kế ở bài học sau.

4.3. Lấy ngẫu nhiên

Khi chọn lệnh phần mềm sẽ giúp bạn tự chọn bất kì một phông nền, nhân vật hay ngoại hình tương ứng của nhân vật.

4.4. Tải đối tượng từ máy tính

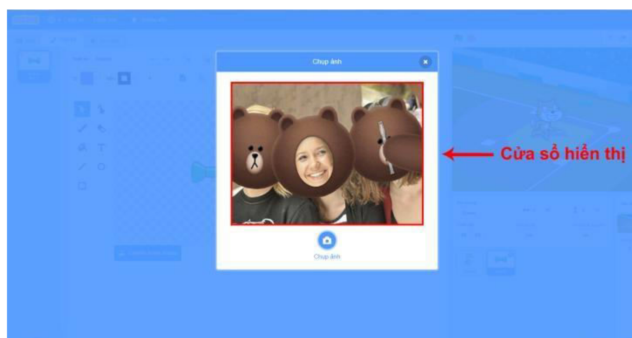
Nếu bạn muốn thêm một đối tượng có sẵn từ máy tính, chọn lệnh để tải từ máy tính. Một cửa sổ xuất hiện để các bạn chọn ảnh hoặc tập tin đối tượng muốn thêm vào chương trình.



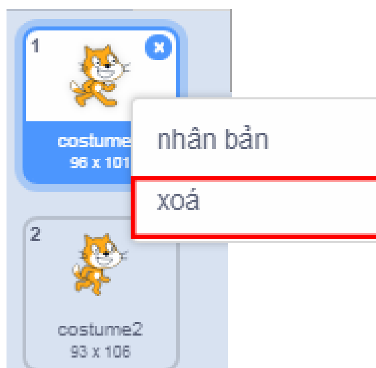
Hình 1.4.5. Cửa sổ thêm đối tượng từ máy tính

4.5. Chụp ảnh

Đặc biệt với việc thêm ngoại hình nhân vật các bạn có thể chọn thêm lệnh để chụp những đối tượng trên màn hình và lưu làm đối tượng mới.



Hình 1.4.6. Chụp ảnh từ máy tính *Lưu ý:* Trong trường hợp muốn xoá phông nền hoặc hình ảnh của đối tượng trong danh sách, chọn phông nền hay hình ảnh muốn xoá và nháy chuột vào dấu hoặc nháy nút phải chuột vào đối tượng muốn xoá và chọn **Xoá**.

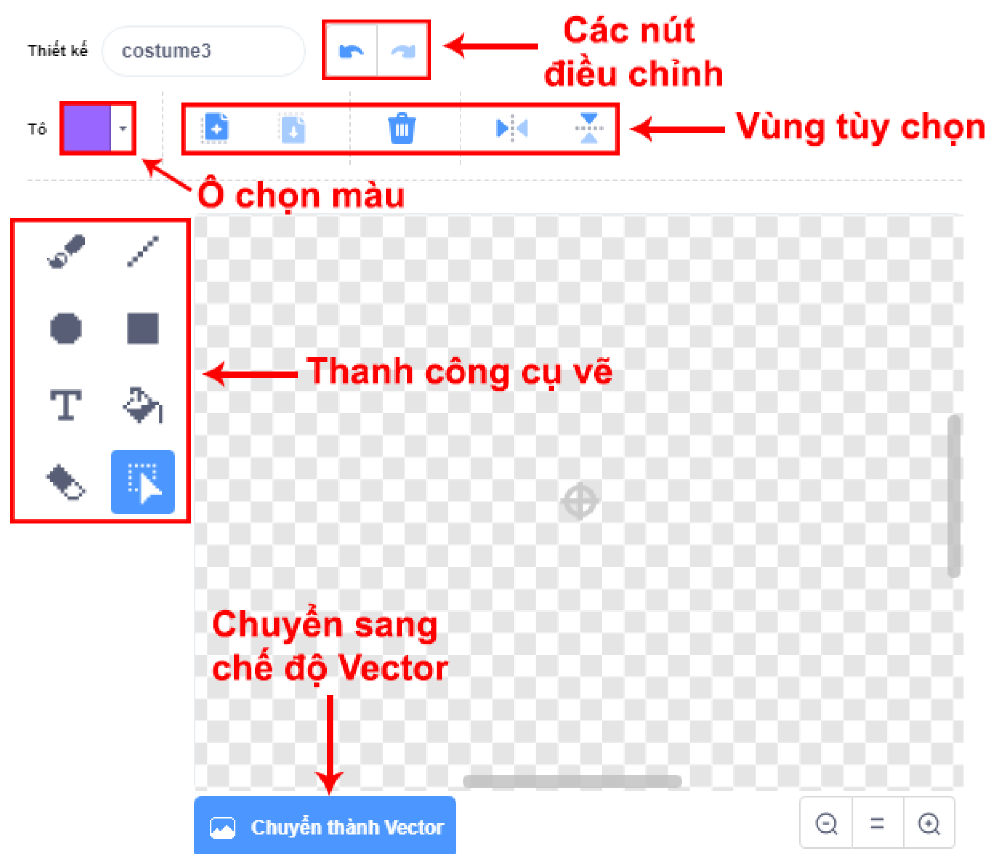


Hình 1.4.7. Xoá đối tượng trong danh sách

BÀI 5 SỬ DỤNG CÔNG CỤ VẼ

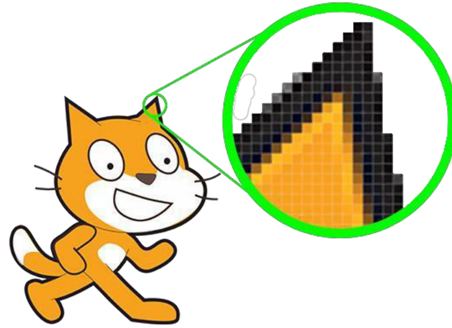
Phông nền hoặc trang phục sẽ được vẽ mới và chỉnh sửa ở Khu thiết kế, nơi có các công cụ vẽ để các bạn vẽ hay chỉnh sửa theo ý muốn. Khu thiết kế có hai chế độ vẽ: Chế độ Bitmap và chế độ Vector.

5.1. Chế độBitmap



Hình 1.5.1. Chế độ Bitmap

Ở chế độ Bitmap, các hình được tạo nên từ những chấm màu gọi là điểm ảnh (tiếng Anh: pixel).



Hình 1.5.2. Các điểm ảnh ở chế độ Bitmap

Khi phóng to ảnh, chất lượng ảnh sẽ bị giảm, ảnh sẽ nhoè đi. Hiện tượng này được gọi là vỡ ảnh.



Hình 1.5.3. Hiện tượng vỡ ảnh

Các bạn có thể thấy ở chế độ này, khu thiết kế gồm các thành phần:

1. Các nút điều khiển

Các nút điều khiển ở bên trên khu thiết kế với các chức năng:

 Huỷ bỏ thao tác vừa thực hiện.

 Thực hiện lại thao tác vừa mới bị hủy.

2. Ô chọn màu

Để có thể chọn được màu cho bút vẽ hay các công cụ vẽ hình, đoạn thẳng,…các bạn có thể sử dụng ô màu có sẵn hoặc dùng công cụ lấy màu.

Ô chọn màu có sẵn bao gồm ba thanh thuộc tính: Thanh màu, Thanh chỉnh độ đậm nhạt và Thanh chỉnh độ sáng. Ô chọn màu có sẵn có 100 màu để lựa chọn, mỗi màu tương ứng với 1 con số.



Hình 1.5.4. Thanh thuộc tính trong ô chọn màu và công cụ lấy màu

Trong trường hợp muốn chọn một màu trên hình vẽ trong Khu thiết kế, các bạn nháy chuột vào



công cụ (góc dưới bên phải ô chọn màu), di chuyển con trỏ chuột vànháy vào vùng có màu muốn lấy.

Hình 1.5.5. Công cụ lấy màu

3. Thanh công cụ vẽ

Thanh công cụ vẽ gồm các công cụ:

- Vẽ một hình bất kì.
- Vẽ đoạn thẳng theo ý muốn.
- Vẽ hình chữ nhật. Khi muốn vẽ hình vuông, nhấn giữ phím **Shift** trong khi kéo thả chuột.
- Vẽ hình elip. Khi muốn vẽ hình tròn, nhấn giữ phím **Shift** trong khi kéo thả chuột.
- Thêm văn bản (chữ) vào hình vẽ.
- Tô màu một vùng khép kín nhất định, như hình vuông, hình chữ nhật, hình tròn,…
- Tẩy, xoá hình.
- Chọn một vùng tuỳ ý.

Các bạn chú ý rằng nếu muốn thay đổi độ dày nét vẽ hoặc màu vẽ cho hình, các bạn cần phải thay đổi trước khi vẽ. Ở chế độ Bitmap thì hình sau khi vẽ sẽ không thể thay đổi màu sắc hay nét vẽ. Việc thay đổi vị trí cũng rất giới hạn, các bạn chỉ có thể thay đổi ngay sau khi vẽ, nếu chuyển sang vẽ hình mới thì sẽ không thể di chuyển hình cũ.

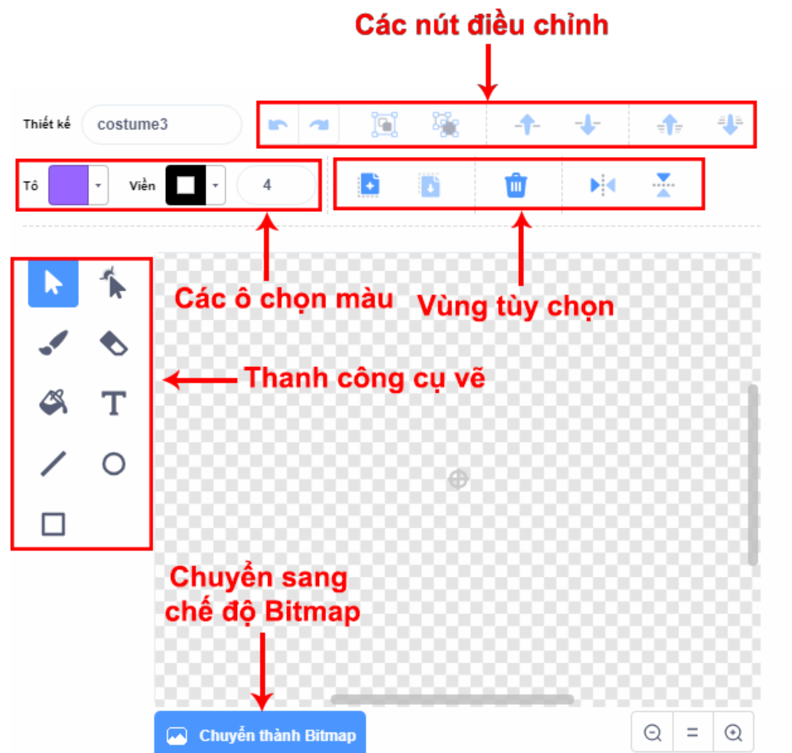
4. Vùng tuỳchọn

Vùng tuỳ chọn ở phía bên trái ô chọn màu. Với mỗi công cụ vẽ khác nhau, vùng này lại hiển thị chức năng tương ứng, ví dụ như thay đổi kích thước nét vẽ, kiểu vẽ, thay đổiphông chữ, sao chép và dán hình ảnh đối tượng, các kiểu lật hình ảnh, …



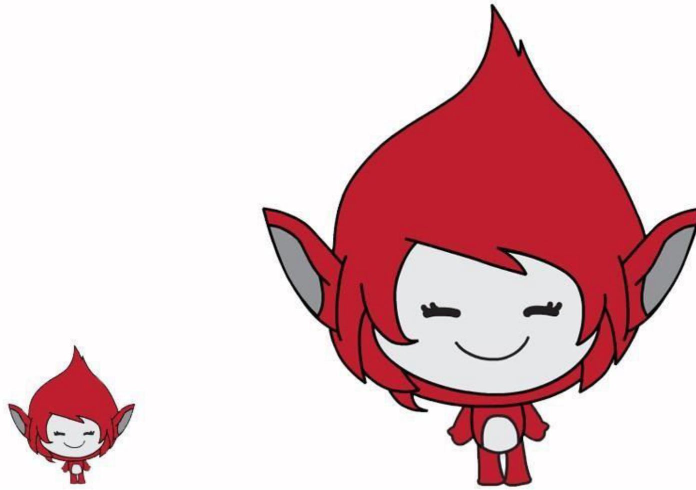
Hình 1.5.6. Các công cụ với vùng tuỳ chọn tương ứng

5.2. Chế độ Vector



Hình 1.5.7. Chế độ Vector

Ở chế độ Vector, khi phóng to ảnh không bị nhoè hay vỡ ảnh mà vẫn rõ ràng sắc nét.



Hình 1.5.8. Phóng to ảnh trong chế độ Vector

Các nút điều khiển, bảng màu và vùng tuỳ chọn trong chế độ Vector đều giống với chế độ Bitmap. Chỉ có thanh công cụ vẽ là khác cả về vị trí và thành phần.

Chọn hình vẽ.

Chỉnh hình, thay đổi hình dáng nét vẽ.

 Vẽ một hình bất kì.

 Vẽ đoạn thẳng theo ý muốn.

 Vẽ hình vuông hoặc hình chữ nhật.

 Vẽ hình elip hoặc hình tròn.

 Thêm văn bản.

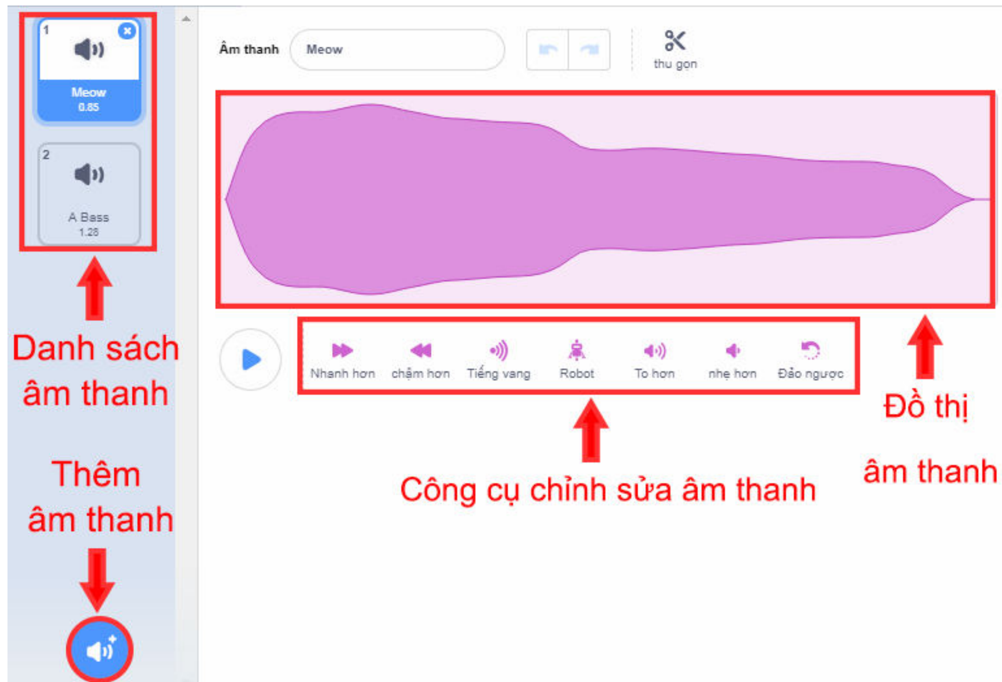
 Tô màu một vùng khép kín nhất định.

 Tẩy, xóa.

Ở chế độ Vector thì các hình vẽ có thể được thay đổi vị trí và kích thước tuỳ ý sau khi vẽ. Dòng văn bản ở chế độ này cũng có thể được thay đổi nội dung và màu sắc bất kì lúc nào.

BÀI 6 KHU ÂM THANH

Chương trình của các bạn sẽ trở nên thú vị hơn rất nhiều nếu có thêm âm thanh. Âm thanh của nhân vật được quản lí trong khu Âm thanh (Sounds).



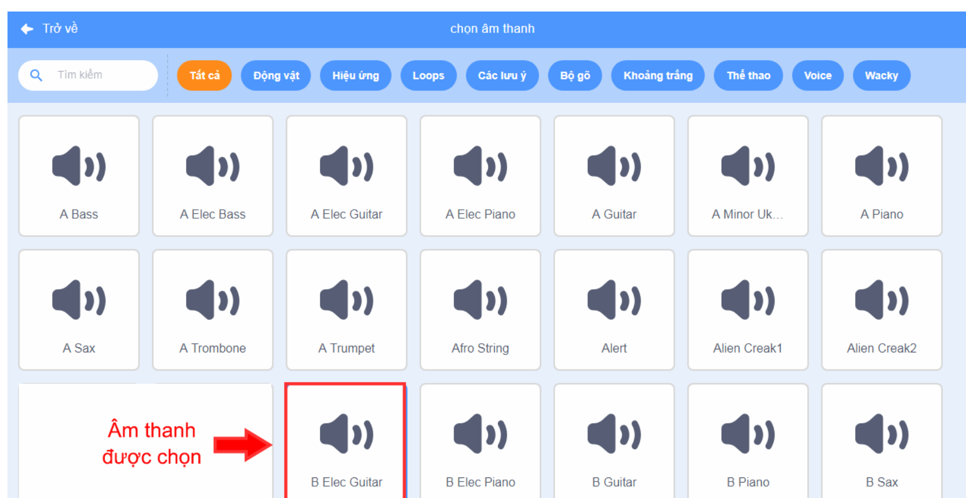
Hình 1.6.1. Khu âm thanh

6.1. Thêm và xoá âm thanh

Để thêm âm thanh cho mỗi đối tượng trong chương trình, chúng ta có ba cách:

1. Chọn từ thư viện

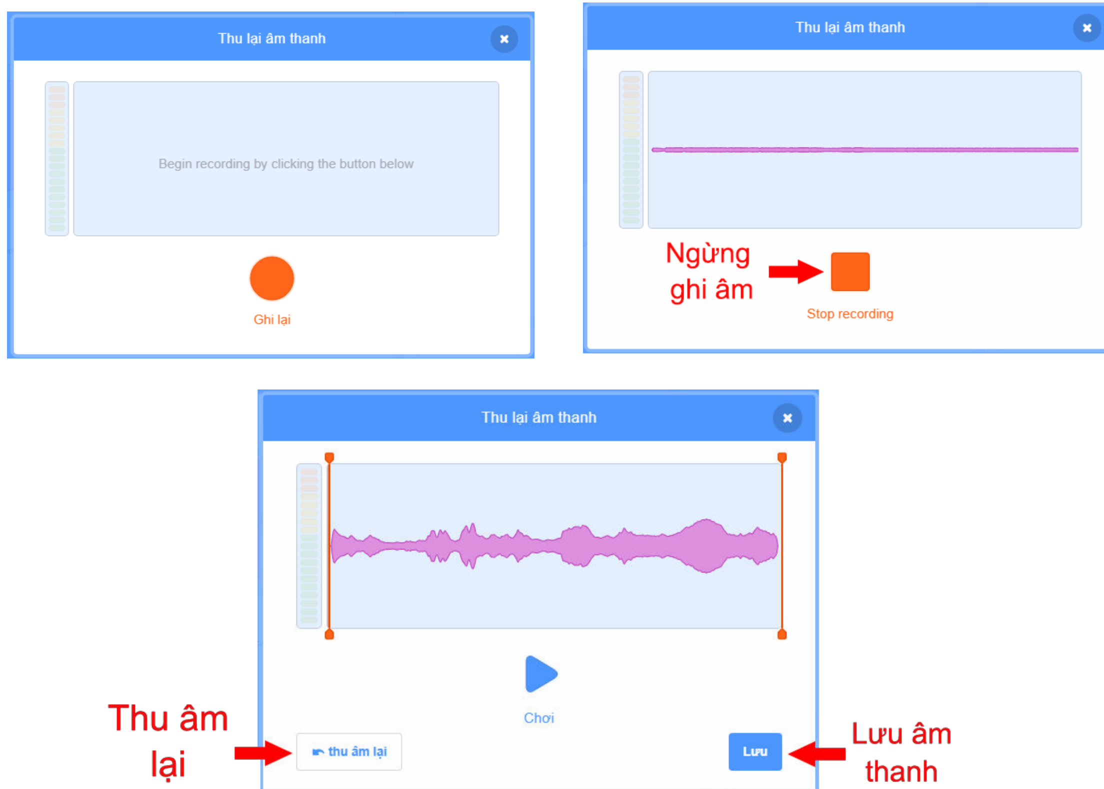
Khi nháy chuột chọn , thư viện âm thanh sẽ xuất hiện. Âm thanh trong thư viện được chia thành nhiều nhóm khác nhau để các bạn có thể tìm kiếm và lựa chọn nhanh chóng.



Hình 1.6.2. Thư viện âm thanh

2. Ghiâm

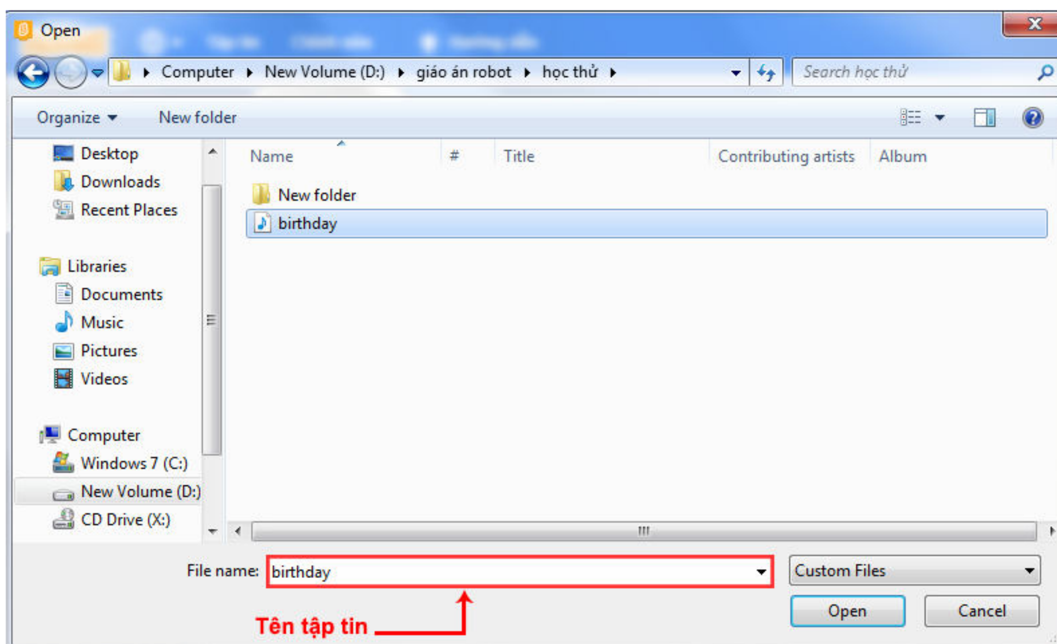
Các bạn có thể tự ghi âm bằng cách chọn lệnh Lúc này một cửa sổ sẽ xuất hiện giúp các bạn thu lại âm thanh qua microphone của máy tính. Các bạn nháy chuột vào nút để tiến hành ghi âm, nháy nút để ngừng việc ghi âm. Sau khi việc thu âm thanh ngừng lại, các bạn có thể nghe lại âm thanh vừa được thu được và lựa chọn Lưu hoặc Thu âm lại.



Hình 1.6.3. Ghi âm

3. Tải từ máy tính

Nếu các bạn muốn thêm âm thanh hay nhạc có sẵn ở trong máy tính,chọn lệnh và một cửa sổ hiện ra cho phép chọn tập tin muốn thêm.



Hình 1.6.4. Cửa sổ thêm âm thanh từ máy tính

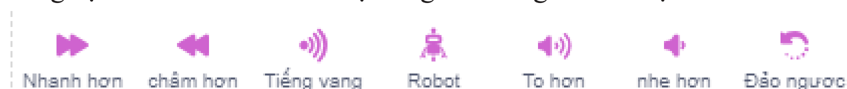
Trong trường hợp các bạn muốn bỏ âm thanh trong danh sách âm thanh, chọn âmthanh muốn bỏ và nháy chuột vào nút hoặc nháy nút phải chuột vào âm thanh muốn bỏ và chọn **Xoá**.



Hình 1.6.5. Xoá âm thanh trong danh sách

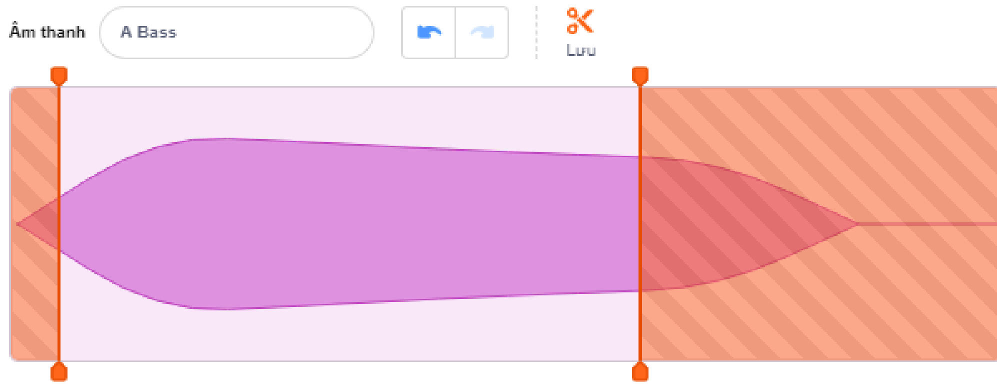
6.2. Chỉnh sửa và thêm hiệu ứng cho âm thanh

Khu âm thanh hỗ trợ rất nhiều thao tác chỉnh sửa cũng như thêm hiệu ứng cho âm thanh. Trong khu này có các công cụ Chỉnh sửa và thêm hiệu ứng để chúng ta làm việc với các đối tượng âm thanh.



Hình 1.6.6. Công cụ chỉnh sửa và thêm hiệu ứng cho âm thanh

. Ngoài ra các bạn có thể rút ngắn âm thanh bằng công cụ 

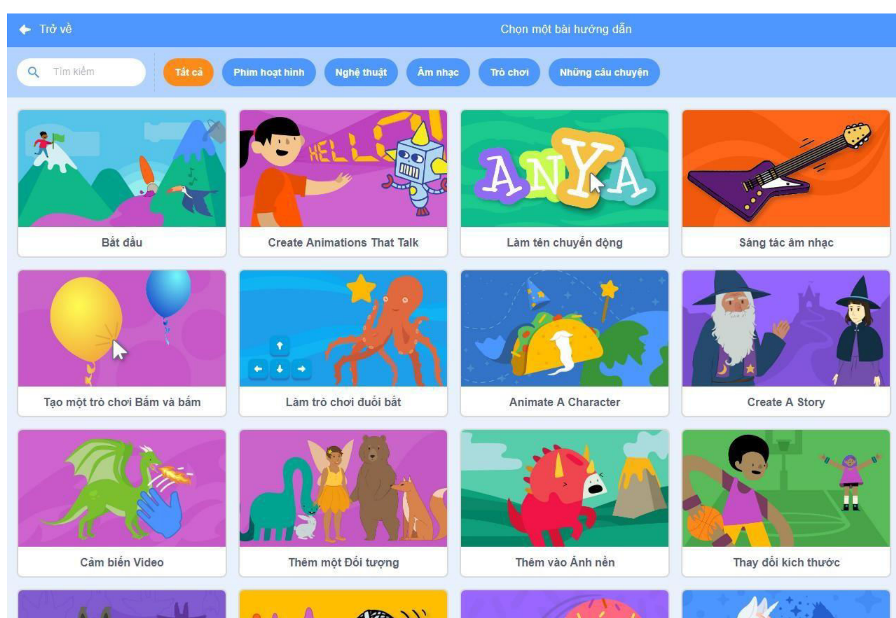


Hình 1.6.7. Thu gọn âm thanh

BÀI 7 THƯ VIỆN HƯỚNG DẪN

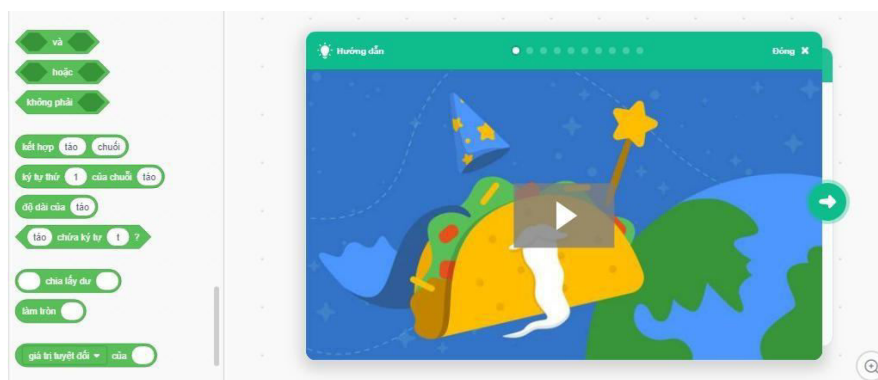
Thư viện hướng dẫn chứa rất nhiều hướng dẫn và ví dụ mẫu. Đây là khu vực rất hữu ích, giúp các bạn hiểu ý nghĩa những khối lệnh, hướng dẫn cách tạo một số hiệu ứng, chuyển động, trò chơi,…

Thư viện hướng dẫn chia làm các chủ đề khác nhau. Với mỗi chủ đề có các ví dụmẫu được thể hiện dưới dạng video hướng dẫn.



Hình 1.7.1. Trang Thư viện hướng dẫn

Chọn một ví dụ bất kì, khi đó video hướng dẫn chi tiết từng bước về ví dụ đó sẽ hiện ra trên khu Lập trình.



Hình 1.7.2. Video hướng dẫn

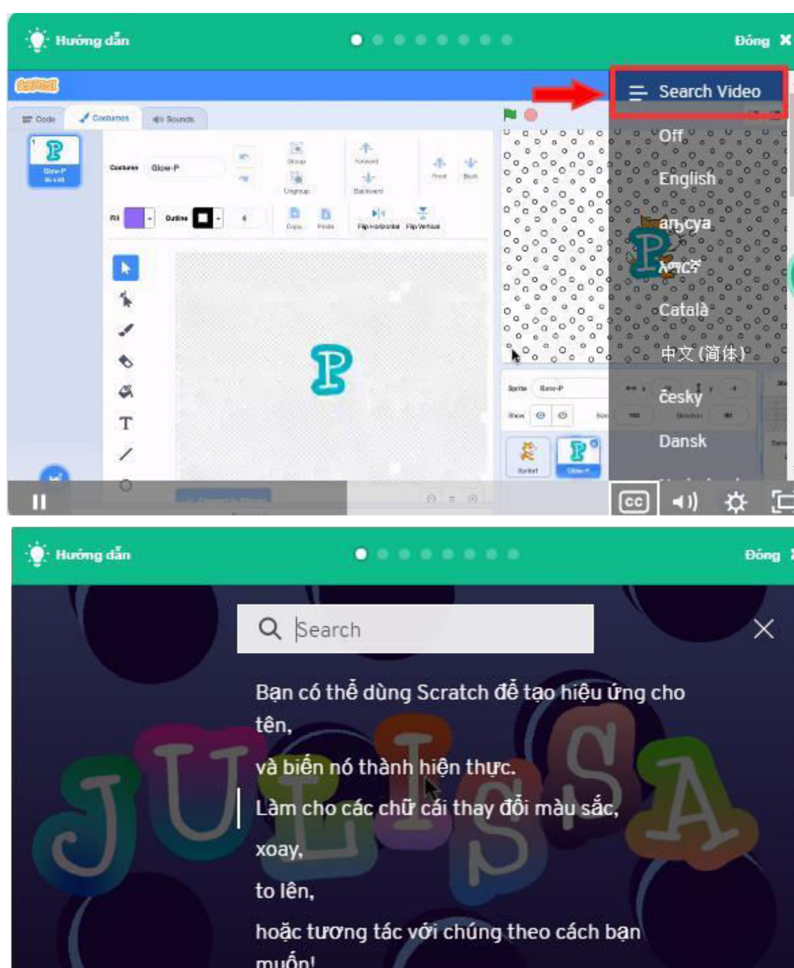
Video này chứa những hướng dẫn các kĩ thuật thiết yếu cho mỗi chương trình, được chia thành:

- **Ý tưởng**: mô tả chương trình mẫu.
- **Thiết kế hình ảnh**: Tạo hình nền, nhân vật cho chương trình,…
- **Lập trình**: Hướng dẫn từng bước lập trình và giải thích các khối lệnh cần dùng.

Tất cả video trong thư viện hướng dẫn đều có phụ đề được bật khi nháy chuột vào nút CC.

Phần phụ đề hiện tại vẫn chưa được hoàn thiện nên sẽ có những video không cóphụ đề Tiếng Việt.

Các bạn có thể xem lại từng câu phụ đề và đoạn video tương ứng bằng cách nhấn vào **Search Video**



Hình 1.7.1. Phụ đề và tra cứu phụ đề

BÀI 8 KHU LẬP TRÌNH

Khu lập trình là nơi chứa các khối lệnh và ghép chúng lại với nhau tạo thành một chương trình theo ý muốn của lập trình viên. Trong bất kì dự án nào, chúng ta đều phải thực hiện các thao tác trên khu vực này.

8.1 Chỉnh sửa và thêm hiệu ứng cho âm thanh

Khu lập trình nằm ở trang **Code**, chứa Bảng danh mục các khối lệnh.



Hình 1.8.1. Bảng danh mục các khối lệnh

Bảng danh mục các khối lệnh gồm các nhóm khối lệnh (gọi tắt là nhóm lệnh) và danh sách các khối lệnh tương ứng với từng nhóm. Scratch 3.0 có hơn 100 khối lệnh cơ bản được chia thành 9 nhóm. Các khối lệnh cùng một nhóm sẽ có màu giống nhau.

Có bốn loại khối lệnh chính trong Scratch 3.0:

1. **Khối lệnh thực hiện:** Có hình dạng với khớp nối ở cả phía trên và phía dưới, dùng để thực hiện một công việc nhất định.

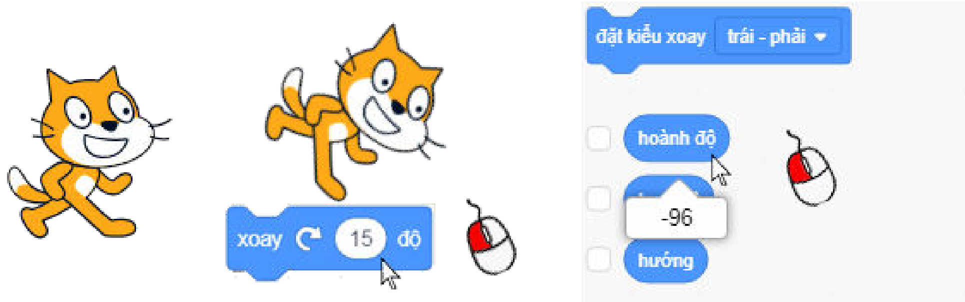
2. **Khối lệnh giá trị:** Có hình dạng hoặc , không có khớp nối nên không thể ghép vào phía trước hoặc phía sau khối lệnh khác được. Chúng lưu trữ một giá trị nhất định, có thể là số, cụm từ/chữ hoặc giá trị Đúng/Sai.

3. **Khối lệnh kích hoạt:** Có hình dạng và chỉ có khớp nối phía dưới. Chúng đợi một sự kiện nào đó (như nhấn phím trên bàn phím hoặc khi nhấp chuột vào) và sẽ thực hiện các khối lệnh bên dưới khi sự kiện đó xảy ra.

4. **Khối lệnh điều khiển:** Có hình dạng . Chúng không những có khớp nối ở phía trên và phía dưới mà còn có khớp nối bên trong, tức là có thể bao lấy một hoặc một đoạn khối lệnh khác bên trong. Thông thường khi chạy một đoạn khối lệnh

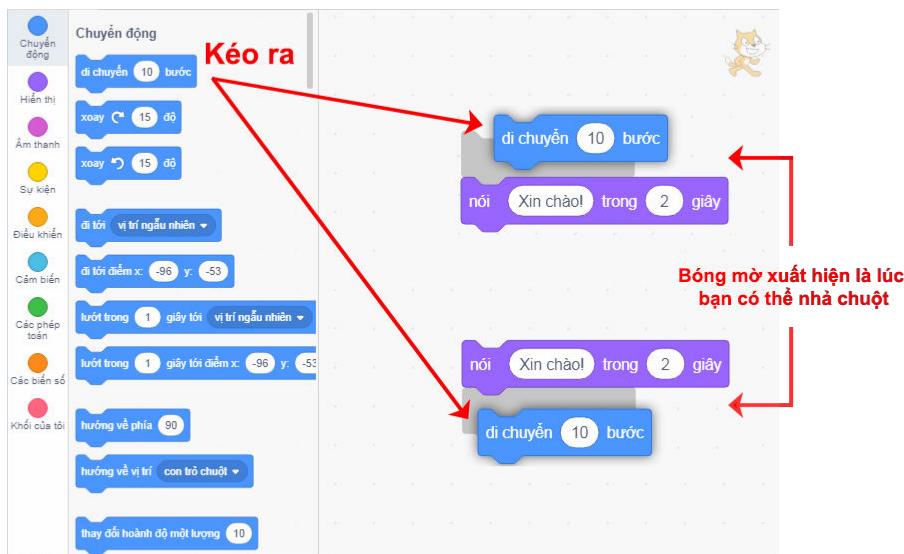
thì các khối lệnh sẽ được thực hiện tuần tự từ trên xuống dưới. Khối lệnh điều khiển có thể làm thay đổi thứ tự thực hiện này.

Các bạn có thể nháy chuột trực tiếp vào khối lệnh để chạy khối lệnh đó. Khối lệnh sẽ thực hiện công việc sau khi nháy chuột 1 giây. Khối lệnh giá trị sẽ hiển thị cho chúng ta giá trị đang lưu trữ.



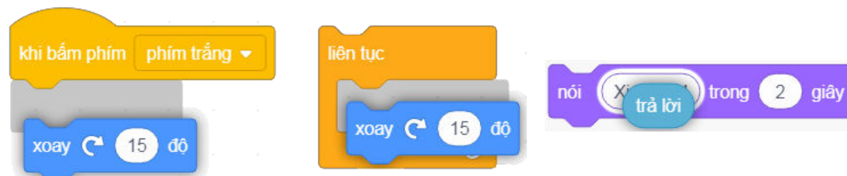
Hình 1.8.2. Thực hiện các khối lệnh bằng cách nháy chuột

Bên cạnh bảng danh mục các khối lệnh là nơi mà các bạn kéo thả các khối lệnh để ghép chúng lại với nhau. Khi ghép các khối lệnh, các bạn cần phải nhấn giữ và kéo chuột cho đến khi giữa hai khối lệnh xuất hiện bóng mờ của khối lệnh hoặc vạch sáng màu trắng thì mới thả chuột. Lúc này hai khối lệnh sẽ được ghép lại với nhau.



Hình 1.8.3. Ghép các khối lệnh thực hiện

Ghép các khối lệnh kích hoạt, khối lệnh sự kiện, khối lệnh giá trị cũng tương tự như vậy .



Hình 1.8.4. Ghép các khối lệnh khác

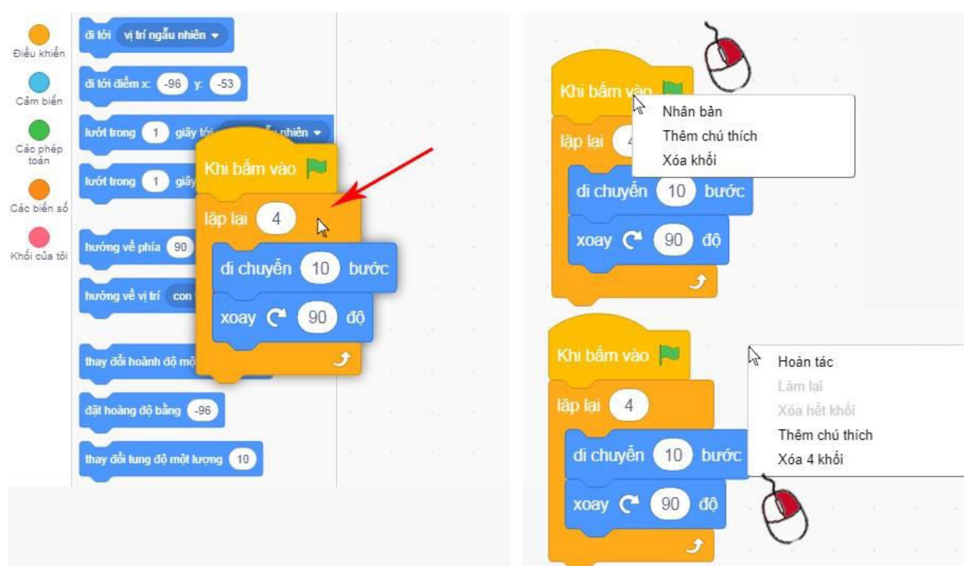
Bằng cách nháy chuột lên trên đoạn khối lệnh các bạn muốn kiểm tra thì các bạn có thể chạy các khối lệnh sau khi ghép mà không cần phải đợi đến khi hoàn thành cả chương trình. Đoạn khối lệnh nào có viền phát sáng tức là nó đang được thực hiện. Muốn đoạn khối lệnh dừng lại thì chỉ cần nháy chuột lại vào đoạn khối lệnh đó.



Hình 1.8.5. Chạy đoạn khối lệnh

Sau khi ghép các đoạn khối lệnh, các bạn có thể tách chúng ra bằng cách nhấn giữ chuột vào khối lệnh cần tách và kéo ra. Khối lệnh các bạn chọn cùng tất cả các khối lệnh theo sau sẽ tách ra khỏi đoạn khối lệnh ban đầu.

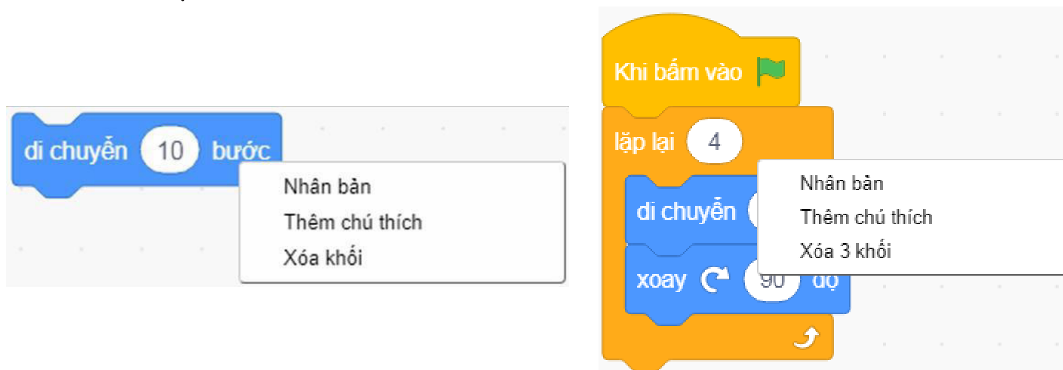
Để xóa khối lệnh, các bạn chỉ cần kéo khối lệnh cần xóa vào khu chứa các khối lệnh trong các nhóm lệnh hoặc các bạn có thể nháy chuột phải vào khối lệnh, khu vực chứa khối lệnh và tùy chọn lệnh Xóa tương ứng.



Hình 1.8.6. Xóa đoạn khối lệnh

Khi nháy nút phải chuột vào một hoặc một đoạn khối lệnh hoặc 1 đoạn bất kì ở trong khu lập

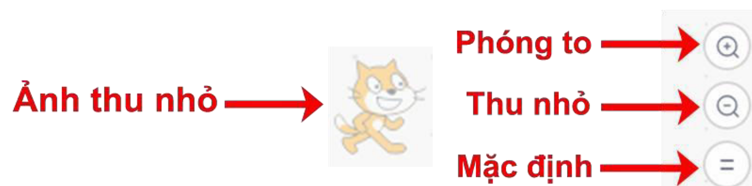
trình, các bạn cũng có một số lựa chọn, chẳng hạn chọn **Thêm chú thích** để viết thêm nhận xét mới hoặc chú thích cho lệnh.



Hình 1.8.7. Các lựa chọn khi nháy nút phải chuột vào khối lệnh

- **Nhân bản:** tạo ra một bản sao của khối lệnh và đoạn khối lệnh gắn theo sau. Bản sao này sẽ dính vào con trỏ chuột để các bạn di chuyển cho đến khi nháy chuột tại vị trí bất kì trên khu lập trình, thậm chí là sang nhân vật hay phông nền khác.
- **Thêm chú thích:** thêm chú thích, nhận xét cho khối lệnh.
- **Xoá khối:** xoá khối lệnh và đoạn khối lệnh theo sau.

Ở góc trên bên phải khu lập trình là ảnh thu nhỏ và toạ độ của nhân vật, giúp các bạn theo dõi dễ dàng hơn, còn ở góc dưới bên phải là công cụ để phóng to, thu nhỏ cáckhối lệnh.



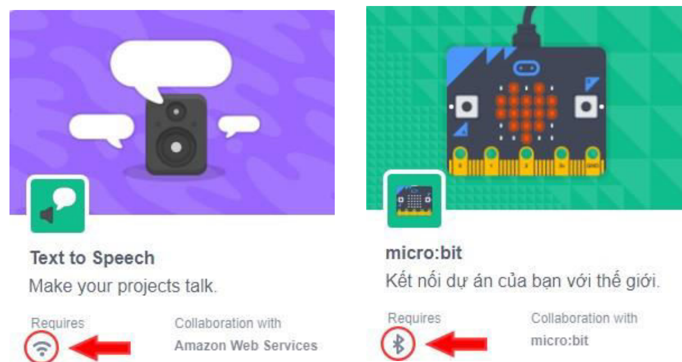
Hình 1.8.8. Công cụ Khu lập trình

8.2 Phần Lập Trình mở rộng

Ngoài 9 nhóm lệnh cơ bản, Scratch 3.0 còn có phần lập trình mở rộng cũng chứa 9 nhóm lệnh nâng cao bao gồm:

- **Âm nhạc**: Chứa các khối lệnh chơi trống, thay đôi nhịp độ và nhạc cụ.
- **Bút vẽ**: Chứa các khối lệnh phục vụ cho các chương trình vẽ hình, in hình, ...
- **Cảm biến**: Các khối lệnh sẽ giúp bạn tạo những trò chơi tương tác với Camera máy tính rất thú vị.
- **Text to Speech**: Khối lệnh lập trình Chuyển văn bản thành giọng nói.
- **Dịch**: Dịch văn bản sang nhiều ngôn ngữ khác nhau.
- **Các nhóm lệnh kết nối với các thiết bị phần cứng**: Makey Makey, micro: bit, LEGO MINDSTORMS EV3, LEGO WeDo 2.0.

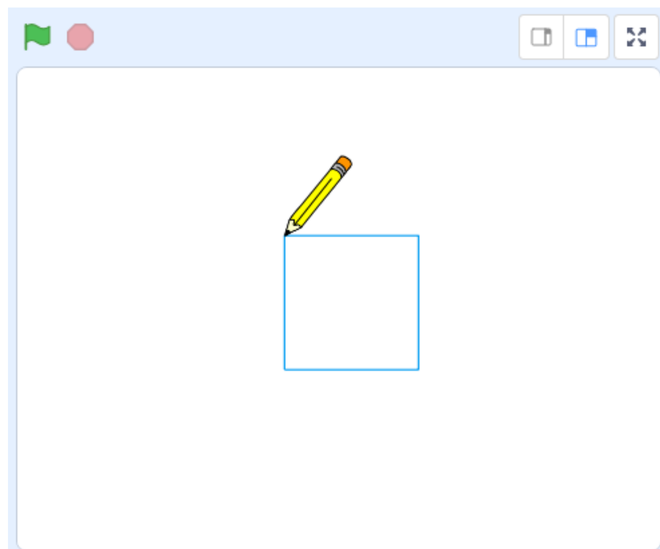
Một số nhóm lệnh mở rộng sẽ có yêu cầu bắt buộc để chương trình có thể hoạt động.



Hình 1.8.9. Yêu cầu của từng nhóm lệnh mở rộng

BÀI 9 VÍ DỤ CƠ BẢN

Sau đây sẽ là một ví dụ vẽ hình vuông đơn giản. Chúng ta cần điều khiển nhân vật chiếc bút để vẽ được một hình vuông như trong *Hình 1.9.1*.



Hình 1.9.1. Ví dụ vẽ hình vuông

9.1. Thêm mới nhân vật

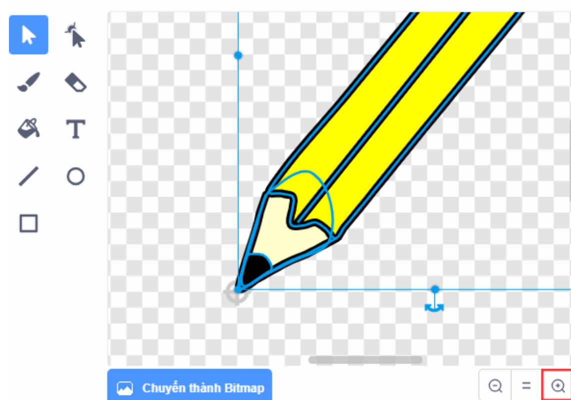
Để thay đổi nhân vật ngầm định là chú mèo thành nhân vật chiếc bút, chúng ta cần xoá chú mèo đi và thêm chiếc bút từ thư viện.



Hình 1.9.2. Thay đổi nhân vật

9.2. Thay đổi tâm nhân vật

Khi các bạn vẽ, nét bút sẽ xuất hiện từ tâm của nhân vật. Ngầm định tâm của nhân vật sẽ ở chính giữa, vì vậy ta cần thay đổi lại vị trí tâm của chiếc bút lên đầu bút để nét vẽ có thể xuất hiện trên đầu bút bằng cách kéo hình dạng bút bên khu thiết kế sao cho đầu bút chì trùng với tâm trang vẽ. Các bạn nên phóng to trang vẽ để xác định tâm được dễ dàng hơn.

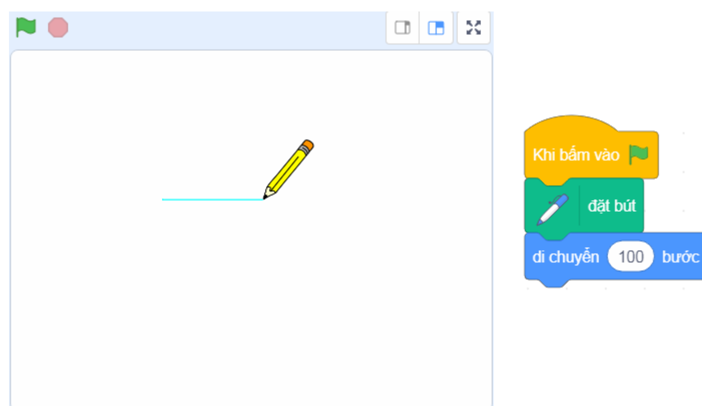


Hình 1.9.3. Thay đổi tâm của nhân vật

9.3. Vẽ cạnh đầu tiên

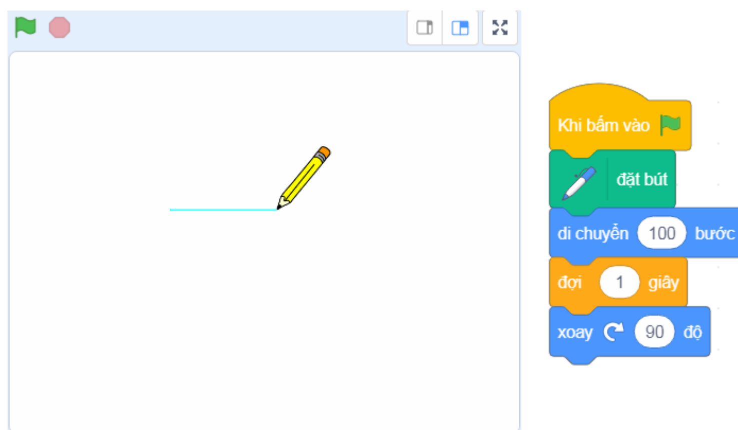
Đầu tiên, các bạn cần khối lệnh Khi bấm vào và đặt bút. Sau khối lệnh này, việc di chuyển của nhân vật sẽ kèm theo nét vẽ xuất hiện từ tâm của nhân vật. Tiếp theo các bạn sử dụng khối lệnh di chuyển 10 bước thay đổi thành 100 bước để vẽ cạnh đầu tiên bằng cách nháy chuột vào ô 10 sau đó sửa thành 100.

Lưu ý: Chọn khối lệnh theo màu sắc của nhóm lệnh.



Hình 1.9.4. Vẽ cạnh hình vuông đầu tiên

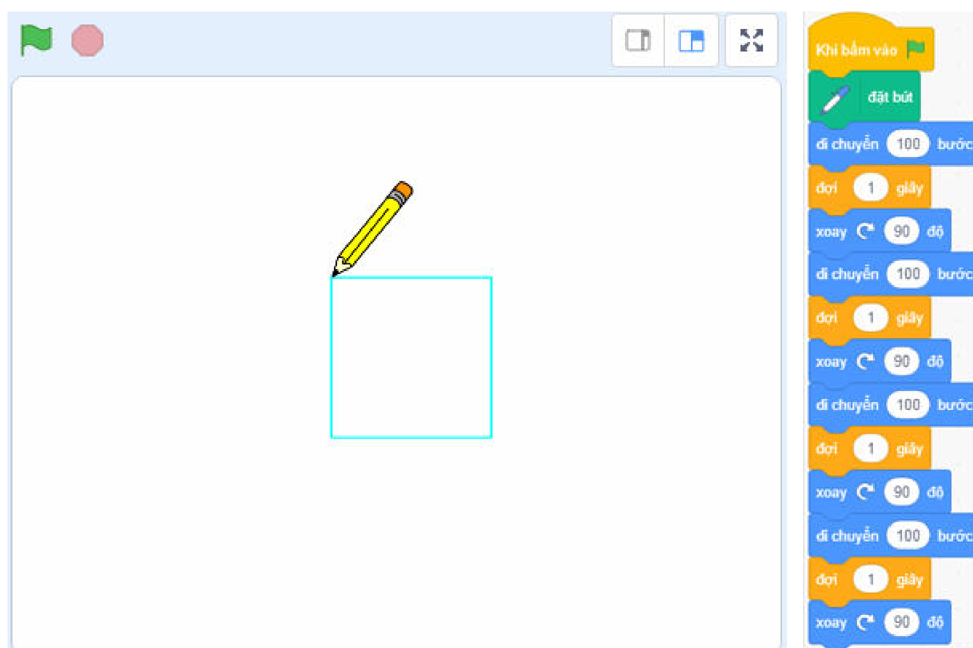
Sau khi vẽ cạnh đầu tiên, các bạn cần đổi hướng chiếc bút xoay xuống dưới mộtgóc vuông bằng 90° để vẽ cạnh tiếp theo bằng khối lệnh xoay ↻ 90 độ. Tuy nhiên, để quan sát chiếc bút di chuyển và xoay dễ dàng hơn, các bạn thêm khối lệnh đợi 1 giây vào giữa hai khối lệnh.



Hình 1.9.5. Khối lệnh xoay và đợi

9.4. Hoàn thành hình vuông

Thực hiện việc di chuyển và xoay để vẽ ba cạnh còn lại giống như cạnh đầu tiên.



Hình 1.9.6. Hoàn thành hình vuông

CHƯƠNG II
Thiệp sinh nhật

Trong chương này, các bạn sẽ được hướng dẫn tạo ra một tấm thiệp sinh nhật đầymàu sắc và thật sinh động. Chúng ta sẽ tìm hiểu cách để thiết kế nhân vật, di chuyển nhân vật, thay đổi màu sắc, thêm âm thanh cho sinh động,… qua đó bước đầu hiểu được một số kiến thức lập trình cơ bản như thực hiện khối lệnh tuần tự, thông báo, sựkiện, cấu trúc vòng lặp,… Sau khi kết thúc chương, các bạn sẽ có thể tự tạo cho mình một tấm thiệp sinh nhật với các nhân vật, hiệu ứng, âm thanh tuỳ ý.

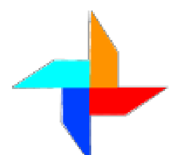
Tổng quan chương trình



Mọi chương trình đều được tạo nên bởi các **nhân vật** (hay còn gọi là **đối tượng**).Lập trình viên sẽ lập trình điều kiển các nhân vật đó thông qua các khối lệnh. Ví dụ làm quả bóng bay, chú chuột di chuyển, dòng chữ đổi màu, chiếc loa phát nhạc,...

Chương trình đầu tiên của chúng ta là tấm thiệp sinh nhật, trong đó có các đối tượng:

1. Chong chóng



Chong chóng được lập trình quay liên tục cho tới khi dừng chương trình.

2. Que

Que có vị trí cố định, là đối tượng gắn với chong chóng, quy định vị trí của chong chóng.

3. Bóng bay

Khi chạy chương trình, Bóng bay xuất hiện tại vị trí ngẫu nhiên và bay từ dưới

4. Chữ Happy



Chữ Happy có vị trí cố định trên sân khấu.

5. Chữ Birthday

Chữ Birthday xuất hiện dưới chữ Happy và liên tục thay đổi màu.

6. Nến



Nến được lập trình cháy liên tục cho đến khi dừng chương trình.

7. Mũ

Mũ sẽ di chuyển lên và xuống nhịp nhàng bên trên chữ P thứ 2 của chữ Happy.

8. Pháo



Pháo cũng di chuyển lên và xuống giống Mũ nhưng ngược với Mũ, khi Mũ di chuyển lên thì Pháo di chuyển xuống và ngược lại.

9. Ngôi sao

Ngôi sao sẽ tạo bản sao xung quanh phần trên của thiệp. Tất cả các bản sao này đều có hiệu ứng đổi màu và nhấp nháy.



BÀI 1 TẠO HÌNH NỀN VÀ VẼ CHONG CHÓNG

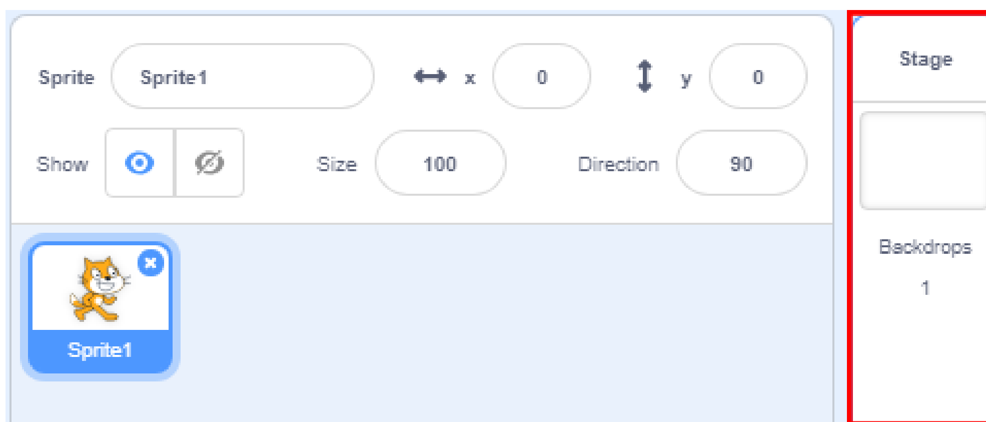
Mở phần mềm Scratch 3.0, tạo và lưu ứng dụng có tên "Thiệp". Trong ứng dụng, trước tiên ta thiết kế hình nền cho thiệp.

1.1. Hình nền



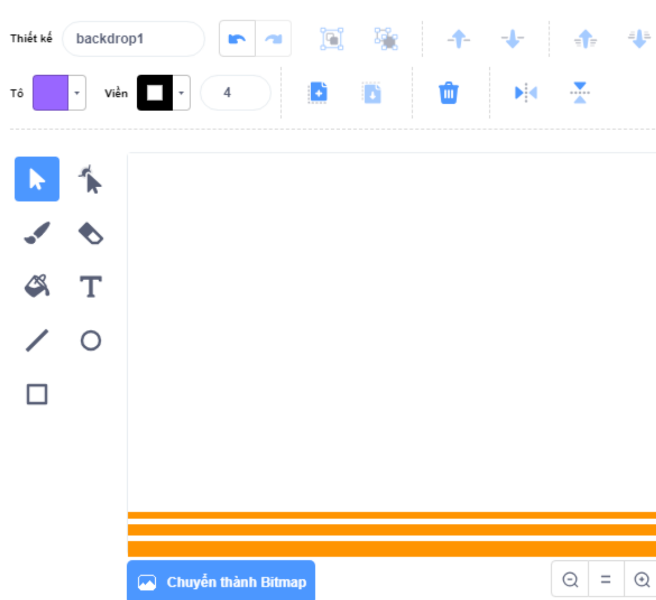
Hình 2.1.1. Hình nền của thiệp

Khi bắt đầu vẽ, ta vào khu thiết kế phông nền bằng cách nháy chuột vào phông nền sân khấu.



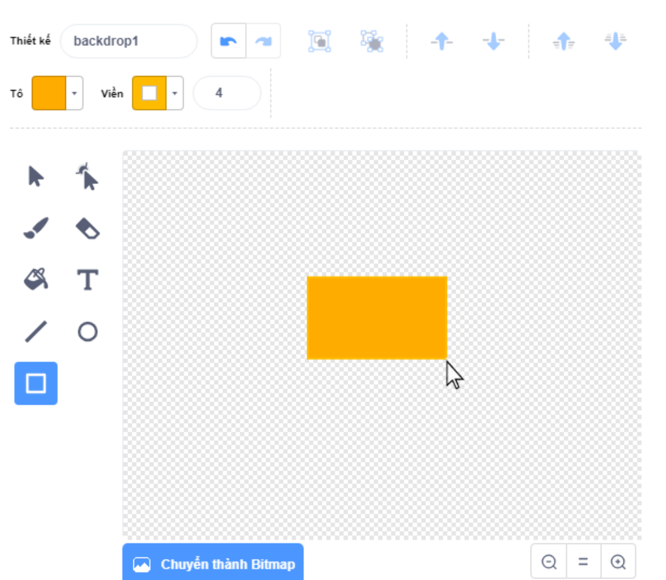
Hình 2.1.2a. Nháy chuột vào phông nền sân khấu

Sau đó nháy chuột vào trang **Phông nền**.



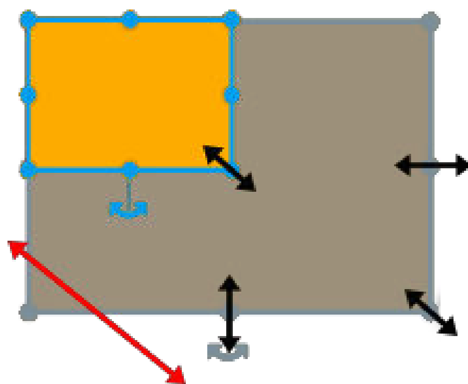
Hình 2.1.2b. Khu thiết kế phông nền

Chúng ta vẽ ba hình chữ nhật ở phía bên dưới bằng công cụ **Vẽ hình chữ nhật** ☐ và chọn màu nền, màu viền. Ta kéo thả chuột trên khu thiết kế để tạo hình chữ nhật.



Hình 2.1.3. Vẽ hình chữ nhật

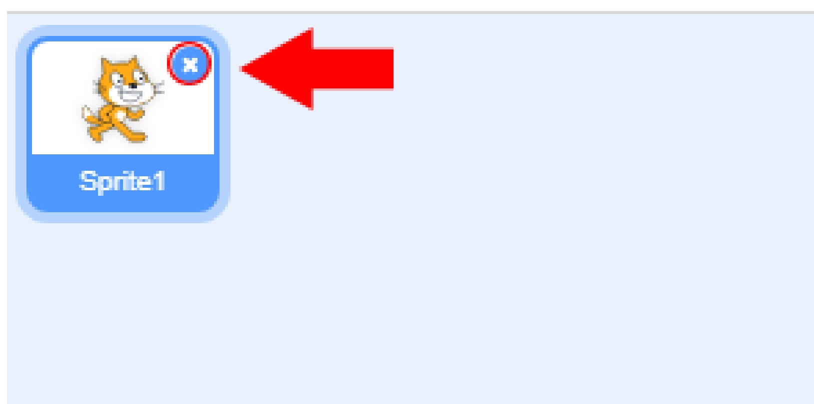
Với hai hình chữ nhật còn lại, ta thực hiện các bước tương tự. Ở chế độ Vector, ta có thể chỉnh sửa lại kích thước, vị trí và màu sắc cho các hình chữ nhật sau khi vẽ xong, còn với chế độ Bitmap thì chỉ có thể thay đổi được màu sắc.



Hình 2.1.4. Chỉnh sửa kích thước hình chữ nhật sau khi vẽ

1.2. Chong chóng

Trước khi tạo các đối tượng cho chương trình, ta xoá đối tượng **Chú mèo** có sẵn bằng cách nháy chuột trái vào biểu tượng góc trên bên phải hình ảnh chú mèo thu nhỏ.



Hình 2.1.5. Xoá đối tượng

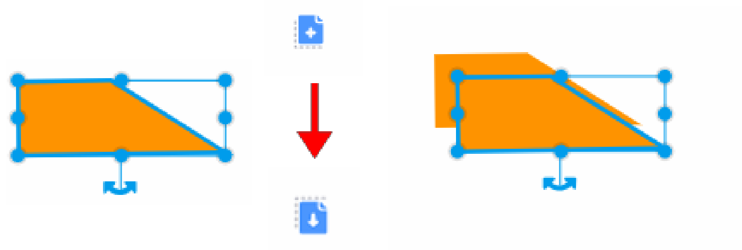
Ở biểu tượng thêm nhân vật mới , ta nháy chuột vào nút **Vẽ** . Khi khu ngoại hình xuất hiện, ta bắt đầu vẽ cánh Chong chóng đầu tiên bằng cách vẽ hình chữ nhật sau đó

chọn công cụ **Chỉnh hình** rồi kéo một góc của hình chữ nhật vào.



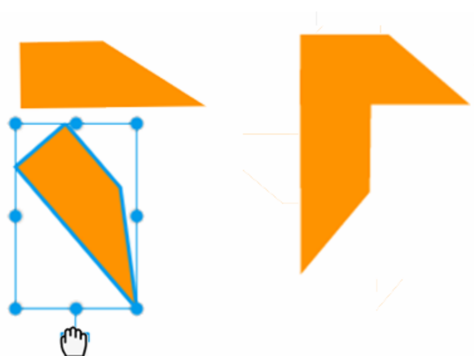
Hình 2.1.6. Vẽ cánh Chong chóng

Để vẽ cánh tiếp theo, ta nháy chuột vào cánh Chong chóng, sau đó nháy chuột vào công cụ **Copy** rồi nháy tiếp vào công cụ **Paste** .



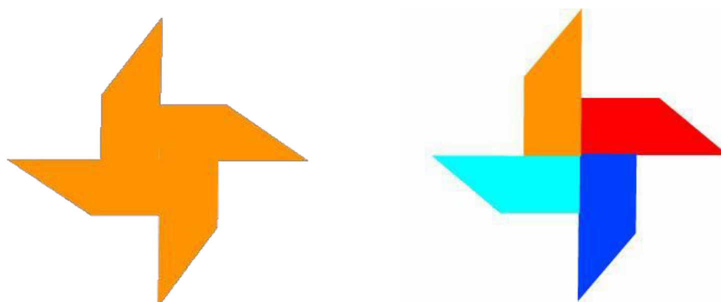
Hình 2.1.7. Tạo bản sao cho cánh Chong chóng

Tiếp theo xoay cánh thứ hai vuông góc với cánh thứ nhất và kéo vào vị trí thích hợp.



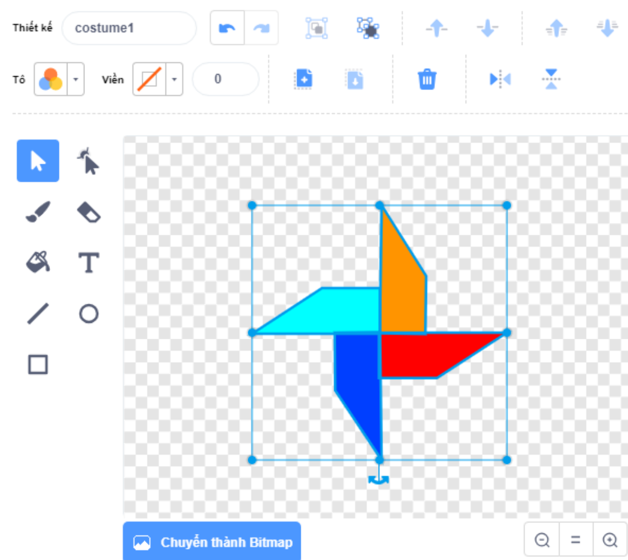
Hình 2.1.8. Vẽ cánh thứ hai

Làm tương tự với hai cánh Chong chóng còn lại. Sau khi vẽ xong, ta đổ màu tuỳ ý cho Chong chóng bằng công cụ **Đổ màu** .



Hình 2.1.9. Đổ màu các cánh Chong chóng

Cuối cùng, chúng ta gộp các cánh lại thành một hình hoàn chỉnh bằng công cụ **Gộp nhóm**



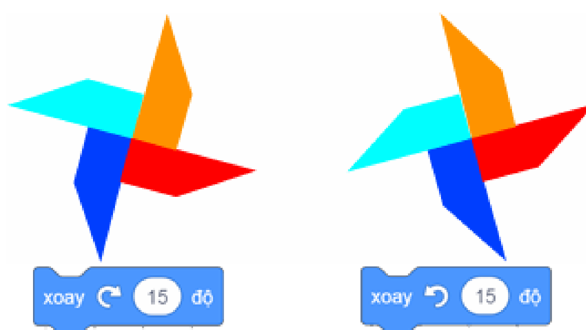
Hình 2.1.10. Chong chóng hoàn chỉnh

BÀI 2 LẬP TRÌNH CHONG CHÓNG QUAY

Sau khi đã tạo được Chong chóng, ta chuyển sang trang **Scripts** để lập trình cho chong chóng quay.

2.1. Khối lệnh lặp và xoay

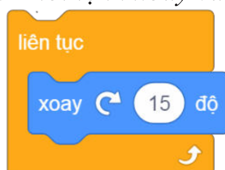
Để lập trình cho Chong chóng quay, ta sử dụng khối lệnh xoay sang phải 15 độ xoay ↻ 15 độ hoặc xoay sang trái 15 độ xoay ↺ 15 độ. Hai khối lệnh này sẽ giúp đối tượng xoay theo chiều sang trái và phải tương ứng với độ lớn góc xoay có thể được thay đổi tuỳ ý.



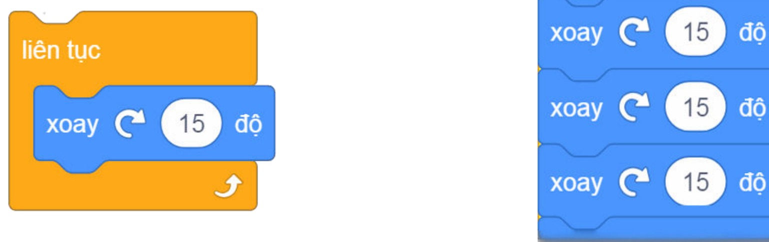
Hình 2.2.1. Khối lệnh xoay

Ta kéo khối lệnh sang khu lập trình và nháy chuột lên khối lệnh, khối lệnh sẽ được chạy. Tuy nhiên, Chong chóng chỉ xoay mỗi lần chúng ta chạy khối lệnh. Nếu muốn Chong chóng xoay liên tục, ta sử dụng khối lệnh lặp để khối lệnh xoay được lặp đi lặp lại. Trong trường hợp này chúng ta sử dụng khối lệnh **Liên tục** trongnhóm **Điều khiển**. Ta kéo khối lệnh **Liên tục** ra sao cho khối lệnh này bao quanh khối lệnh xoay.

Hình 2.2.2. Khối lệnh xoay và khối lệnh lặp



Trong lập trình gọi đây là một vòng lặp, những khối lệnh ở bên trong vòng lặp (được bao quanh bởi khối lệnh lặp) sẽ được thực hiện lặp đi lặp lại nhiều lần. Ở đây chúng ta sử dụng khối lệnh **Liên tục** nên khối lệnh xoay ở bên trong sẽ được thực hiện lặp đi lặp lại mãi mãi cho đến khi nút **Dừng lại** được chọn, và kết quả là Chong chóng sẽquay liên tục không ngừng.

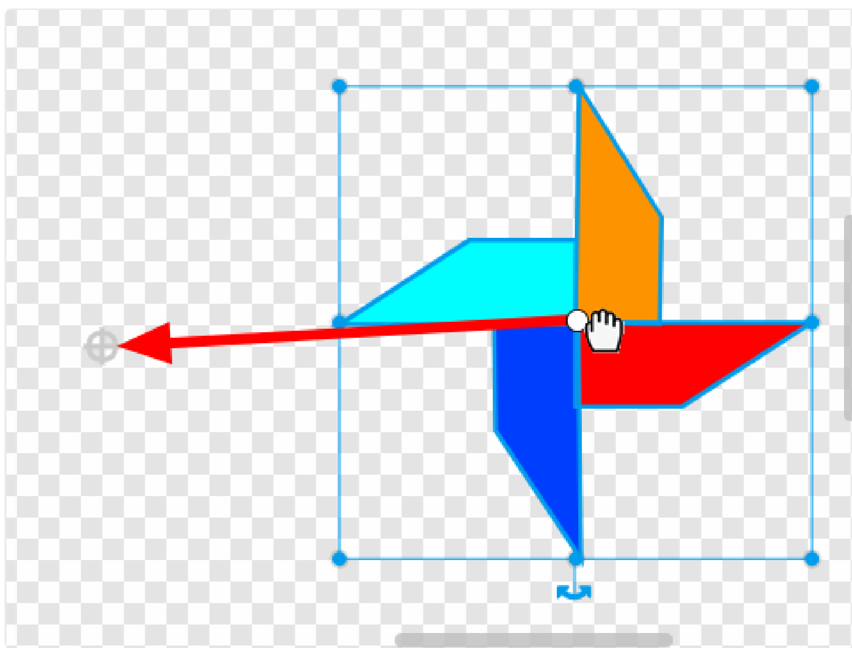


Hình 2.2.3. Hoạt động của vòng lặp

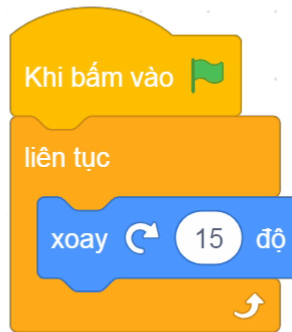
2.2. Điều chỉnh tâm của nhân vật

Khi khối lệnh được chạy, Chong chóng đã quay được nhưng lại quay quanh một điểm không phải điểm chính giữa của Chong chóng. Khi quay, đối tượng thường quay quanh tâm của nó, vậy nên để Chong chóng quay quanh điểm chính giữa của nó, cần đặt lại tâm của Chong chóng vào điểm chính giữa. Ta chuyển sang khu thiết kế, gộp nhóm các cánh của Chong chóng và di chuyển Chong chóng đến vị trí chính giữa. Các bạn có thể phóng to trang thiết kế bằng biểu tượng để dễ dàng xác định tâm.

Hình 2.2.4. Đặt lại tâm cho Chong chóng



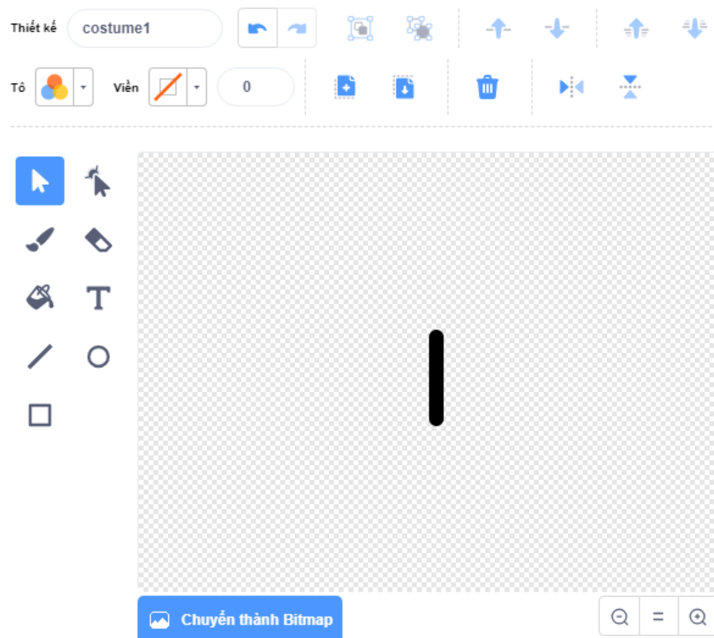
Cuối cùng, để khối lệnh chạy khi nháy chuột vào lá cờ xanh, ta sử dụng khối lệnh **Khi nhấn vào lá cờ xanh** và ghép lên trên khối lệnh.



Hình 2.2.5. Đoạn khối lệnh hoàn chỉnh

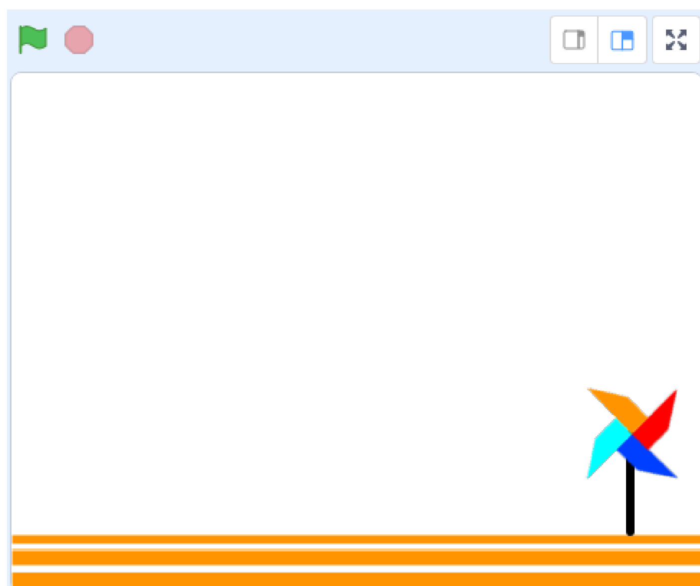
2.3. Đối tượng que

Trong chương trình, Chong chóng được gắn vào một chiếc Que. Nếu vẽ Que trực tiếp lên hình nền thì sẽ không thể di chuyển Que trong trường hợp muốn Chong chóng và Que thay đổi vị trí, còn nếu vẽ Que trực tiếp vào hình ảnh nhân vật của Chong chóng thì khi Chong chóng quay chiếc Que sẽ quay theo. Chính vì vậy, Que là một đối tượng tách riêng ra khỏi Chong chóng. Ta tạo nhân vật mới bằng cách dùng công cụ **Vẽ** , sau đó vẽ Que bằng hình chữ nhật.



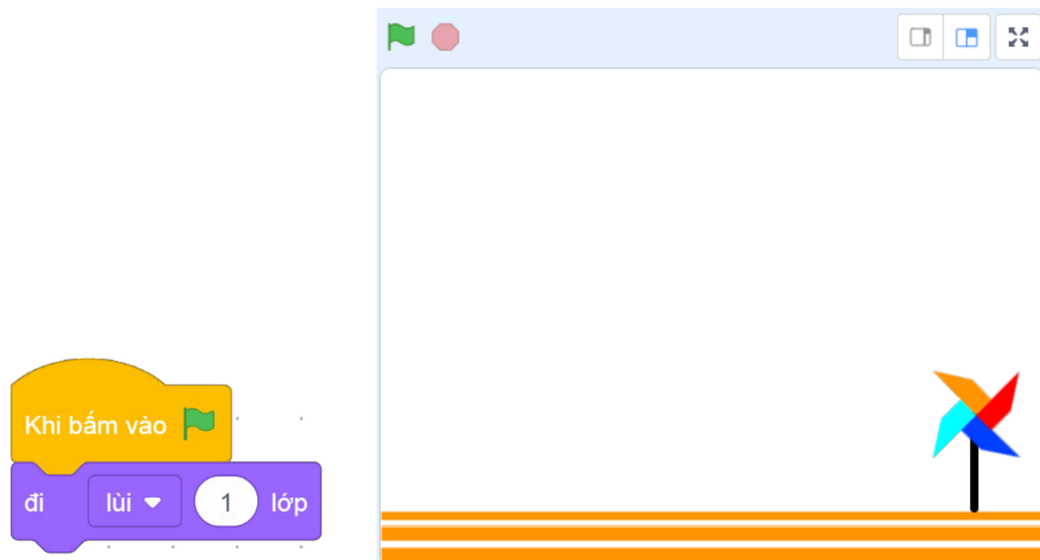
Hình 2.2.6. Vẽ đối tượng Que

Sau khi chỉnh tâm cho Que, các bạn kéo Que đến vị trí chong chóng sẽ thấy Que đang nằm phía trước và che khuất Chong chóng.



Hình 2.2.7. Vị trí của Que đứng trước Chong chóng trên sân khấu

Các nhân vật trên sân khấu đứng trước hay đứng sau là do được sắp xếp theo cáclớp từ trước đến sau và mỗi nhân vật ở một lớp khác nhau. Để đặt Que ra phía sau Chong chóng, ta sử dụng khối lệnh **Đi lùi 1 lớp** đi lùi 1 lớp.



Hình 2.2.8. Đặt vị trí của Que ra sau Chong chóng

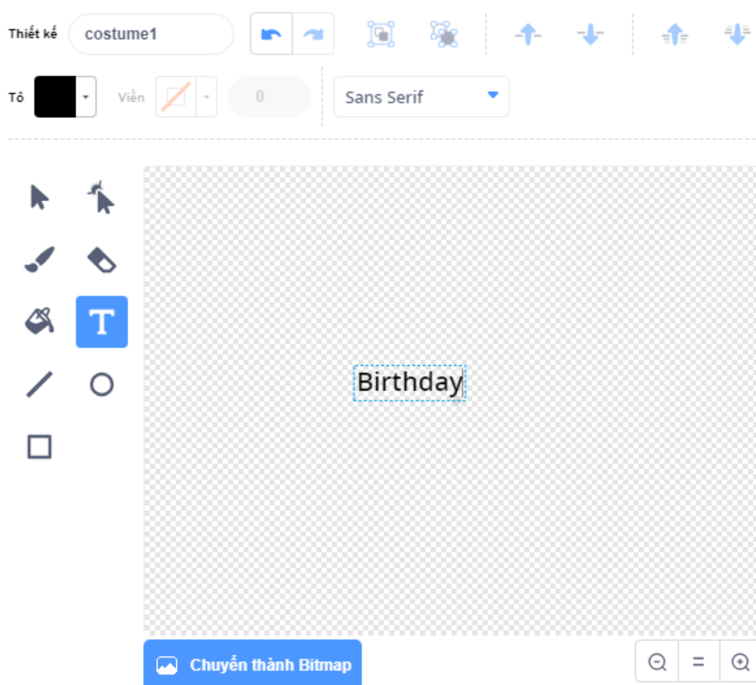
BÀI 3 LẬP TRÌNH ĐỐI TƯỢNG CHỮ

Ở bài học này, các bạn sẽ được học cách thêm chữ Birthday và lập trình thay đổi hiệu ứng cho chữ ở trên thiệp.

3.1. Thêm và chỉnh sửa chữ

Chữ Birthday cũng là một đối tượng. Ta nháy vào nút **Vẽ** trong biểu tượng thêm nhân vật để thêm một nhân vật mới, trong khu thiết kế cho nhân vật mới, chúng ta sẽ thiết kế chữ Birthday.

Để thêm dòng chữ trong khu thiết kế, ta nháy vào nút **Thêm đoạn văn bản** T rồi nháy chuột vào vị trí bất kì trên khu thiết kế và gõ chữ "Birthday".



Hình 2.3.1. Thêm chữ cho thiệp

Thay đổi kích thước chữ bằng cách nháy vào **Nút chọn** , sau đó nhấp vào chữ Birthday để chọn. Sau khi chọn, khung viền bao quanh chữ sẽ xuất hiện, nháy vào góc của khung viền và kéo khung viền to ra hoặc thu nhỏ lại.



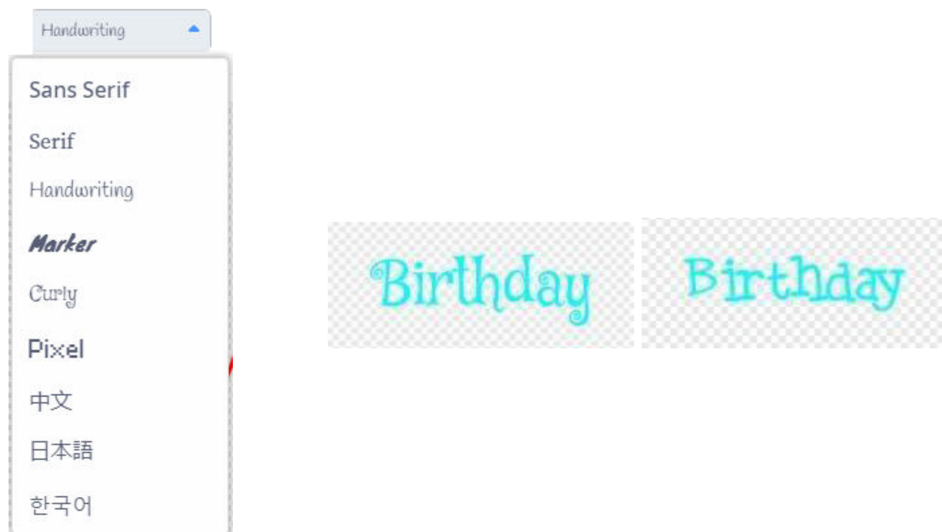
Hình 2.3.2: Thay đổi kích thước chữ

Để đổi màu cho chữ, các bạn chọn màu ở bảng màu rồi sử dụng công cụ **Đổ màu** , hoặc chỉ cần nháy nút **Chọn** , sau đó chọn chữ và chọn màu ở bảng màu là màu chữ sẽ tự động thay đổi theo.



Hình 2.3.3: Thay đổi màu

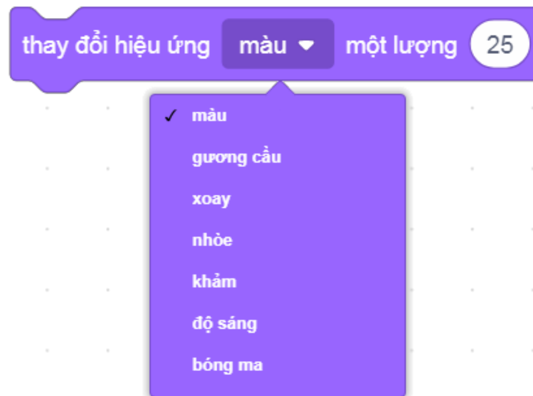
Để thay đổi nội dung hoặc phông chữ, ta nháy vào nút , sau đó nháy vào chữ muốn chỉnh sửa. Lúc này, ta có thể thay đổi nội dung của chữ hoặc thay đổi phông chữ bằng cách chọn phông chữ mình muốn trong danh sách phông chữ.



Hình 2.3.4. Thay đổi phông chữ

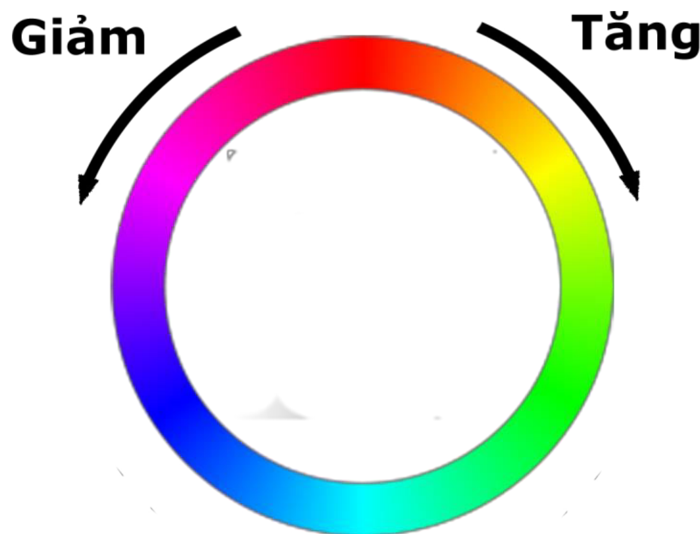
3.2. Hiệu ứng

Việc thay đổi hiệu ứng của chữ được điều khiển bằng khối lệnh **Thay đổi hiệu ứng** ở nhóm **Ngoại hình**. Khối lệnh gồm loại hiệu ứng và lượng thay đổi hiệu ứng dưới dạng số. Trong khối lệnh hỗ trợ thay đổi nhiều hiệu ứng khác nhau.



Hình 2.3.5: Khối lệnh thay đổi hiệu ứng

Trong tấm thiệp, chữ sẽ luôn đổi màu, vì thế các bạn cần chọn hiệu ứng **Màu** trong khối lệnh. Màu chữ thay đổi phụ thuộc vào lượng giá trị trong khối lệnh. Lượng giá trị thay đổi là số dương (tăng) thì màu sẽ thay đổi theo chiều dương của kim đồng hồ, còn nếu lượng giá trị thay đổi là số âm (giảm) thì màu sẽ thay đổi theo chiều ngược lại,như được mô tả như trong hình 2.3.6.



Hình 2.3.6. Màu thay đổi dựa trên lượng giá trị thay đổi

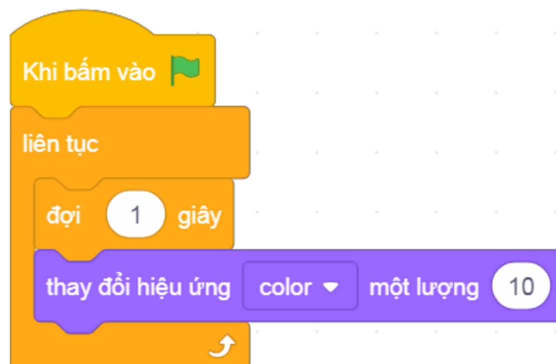
Lượng giá trị thay đổi của hiệu ứng màu có chu kì 200, tức là khi lượng giá trị thay đổi đạt đến một lượng bằng 200 thì hiệu ứng màu sẽ quay trở lại là màu gốc ban đầu. Với màu gốc là màu thì sự thay đổi hiệu ứng màu với giá trị dương từ 0 đến 200 được minh hoạ như trong hình 2.3.7.



Hình 2.3.7. Hiệu ứng màu thay đổi theo chu kì giá trị hiệu ứng là 200

Để chữ có thể đổi màu liên tục, khối lệnh **Thay đổi hiệu ứng màu một lượng**

thay đổi hiệu ứng màu một lượng 25 được thực hiện lặp đi lặp lại nhiều lần, khối lệnh **Liên tục** được sử dụng để chữ đổi màu cho đến khi dừng chương trình. Lúc này chữ đã đổi màu được liên tục nhưng rất nhanh. Ghép khối lệnh đợi đợi 1 giây vào trong vòng lặp để sau mỗi lần đổi màu, chữ sẽ đợi thêm 1 giây rồi mới đổi màu tiếp. Và cuối cùng là khối lệnh **Kích hoạt** để khi bắt đầu chương trình (khi nháy vào nút lệnh) thì chữ sẽ bắt đầu đổi màu.



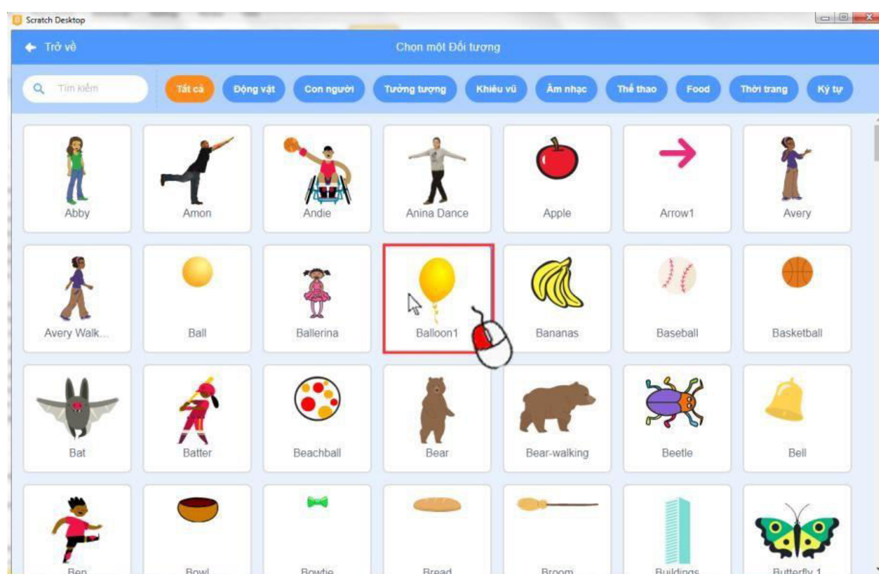
Hình 2.3.8. Đoạn khối lệnh hoàn chỉnh

BÀI 4 LẬP TRÌNH ĐỐI TƯỢNG BÓNG BAY

Trong bài học này, chúng ta sẽ thêm đối tượng Bóng bay và lập trình để quả bóng có thể bay.

4.1. Thêm nhân vật Bóng bay

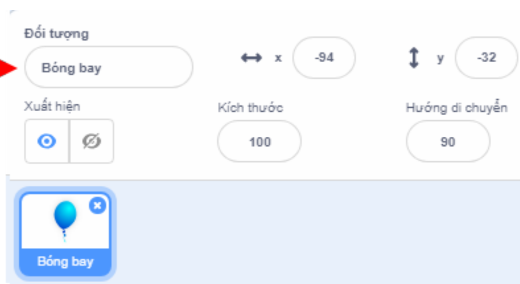
Để thêm quả bóng bay vào chương trình, ta chỉ cần nháy nút ở khung nhân vật để có thể chọn quả bóng bay từ thư viện có sẵn. Sau khi chọn nháy nút OK để hoàn tất.



Hình 2.4.1. Thêm bóng bay từ thư viện

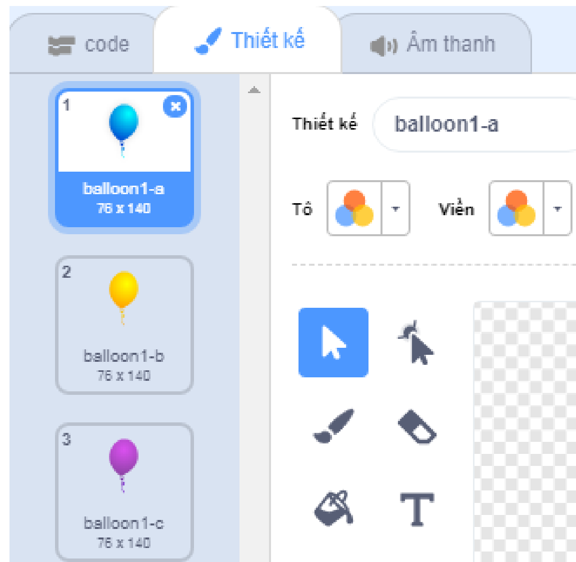
Tên ngầm định của nhân vật khá khó đọc, ta nên thay đổi lại tên để dễ dàng tìm kiếm và không bị nhầm lẫn. Ở danh sách các đối tượng, ta nháy vào biểu tượng ở góc trên bên trái của nhân vật trong khung nhân vật để chỉnh sửa lại tên.

Đổi tên nhân vật



Hình 2.4.2. Đổi tên cho nhân vật

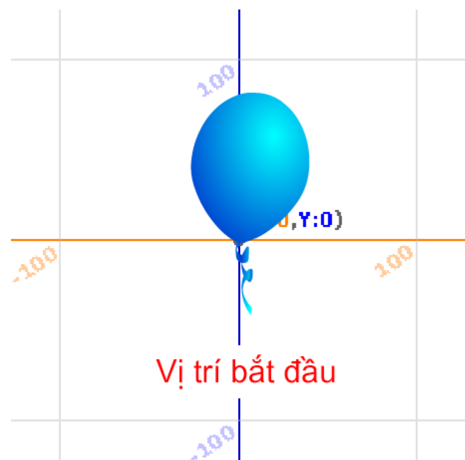
Nhân vật Bóng bay có sẵn ba hình dạng với màu sắc khác nhau, ta thay đổi hình ảnh của nhân vật tuỳ theo sở thích.



Hình 2.4.3. Đổi hình dạng Bóng bay

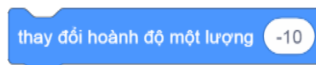
4.2. Lập trình để quả bóng bay

Nhân vật Bóng bay trong tấm thiệp cần được lập trình để có thể bay từ dưới lên trên. Trước tiên, chúng ta sẽ đi tìm hiểu về sự di chuyển của nhân vật trên sân khấu bằng cách thay đổi toạ độ x (hoành độ) và y (tung độ). Ở thẻ **Code**, các bạn chạy lần lượt các khối lệnh thay đổi hoành độ một lượng và thay đổi tung độ một lượng với hai giá trị là 10 và –10, sau đó theodõi kết quả.

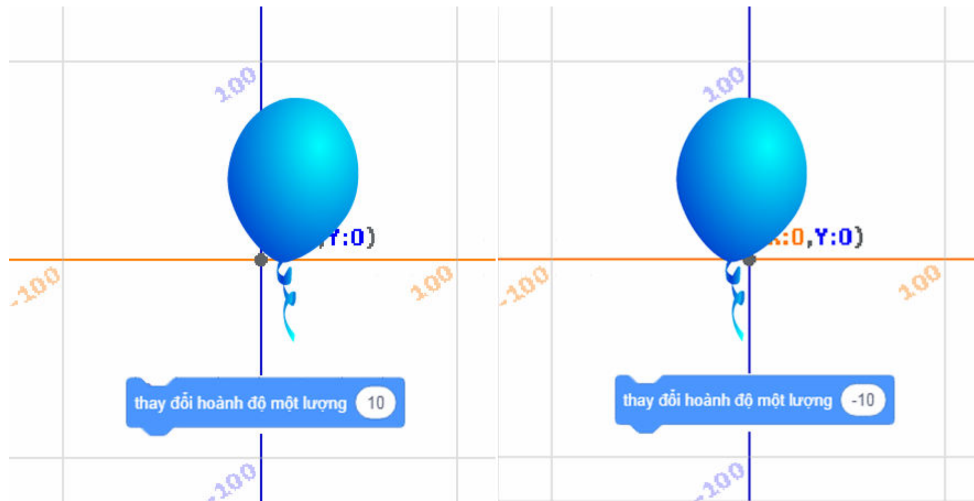


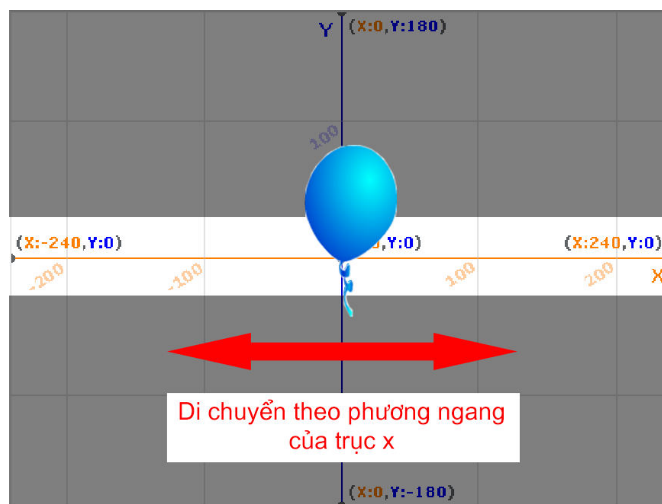
Hình 2.4.4. Vị trí bắt đầu của quả bóng

Khi chạy khối lệnh thay đổi hoành độ một lượng 10, quả bóng di chuyển sang phải 10



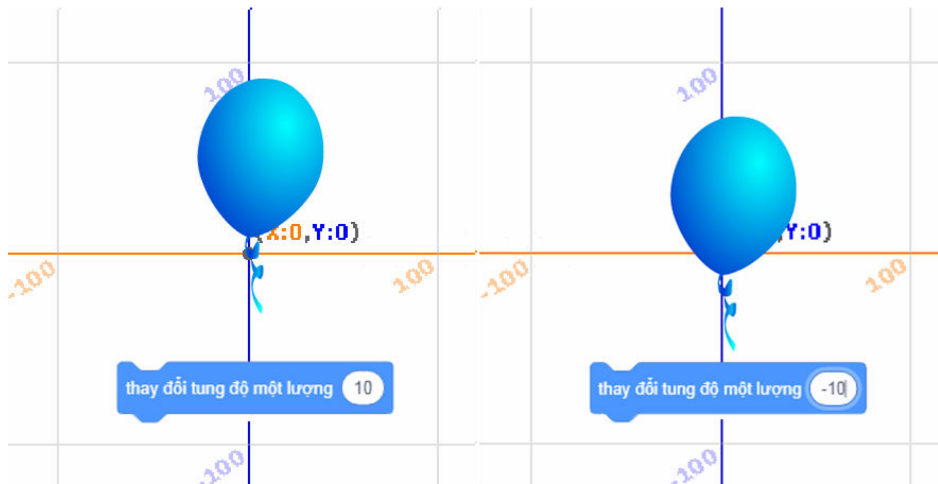
bước (10 pixel), còn chạy khối lệnh sẽ khiến quả bóng di chuyển sang trái 10 bước. Như vậy, việc thay đổi hoành độ của nhân vật sẽ khiến nhân vật di chuyển theo phương ngang, tức là phương theo trục x.

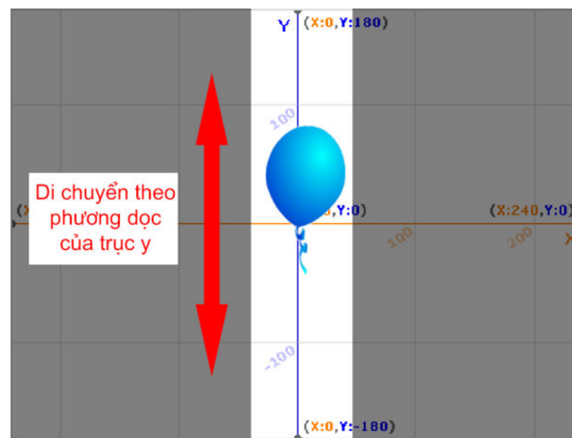




Hình 2.4.5. Thay đổi hoành độ một lượng nhất định

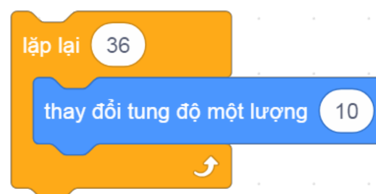
Khi chạy khối lệnh thay đổi tung độ một lượng 10, quả bóng sẽ di chuyển lên trên 10 bước, còn nếu lượng thay đổi là –10 như trong khối lệnh thay đổi tung độ một lượng -10 thì quả bóng sẽ di chuyển xuống dưới 10 bước. Vậy việc thay đổi tung độ của nhân vật sẽ khiến nhân vật di chuyển theo phương thẳng đứng, tức là theo phương của trục y.





Hình 2.4.6. Thay đổi tung độ một lượng nhất định

Quả bóng cần phải di chuyển theo phương thẳng đứng với chiều hướng từ dưới lên trên, vì vậy các bạn sử dụng khối lệnh thay đổi tung độ một lượng ( ). Ở Chương I, chúng ta đã được biết sân khấu có chiều dọc là 360 bước, vậy nên để di chuyển hết chiều này thì bóng bay cũng phải di chuyển hết 360 bước. Trong trường hợp này, chúng ta sẽ điều khiển quả bóng di chuyển 36 lần, mỗi lần 10 bước bằng cách thực hiện khối lệnh thay đổi tung độ một lượng (10) lặp đi lặp lại 36 lần.

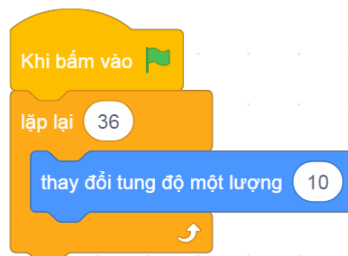


Hình 2.4.7. Đoạn khối lệnh di chuyển của Bóng bay

Ta có thể điều chỉnh số lần lặp để thay đổi độ dài quãng đường di chuyển cũng như điều chỉnh lượng

giá trị y để thay đổi tốc độ di chuyển của quả bóng nhanh hay chậm tuỳ ý.

Cuối cùng, để đoạn khối lệnh chạy khi nháy vào nút lệnh , ta sử dụng khối lệnh **Khi nhấn vào lá cờ xanh** Khi bấm vào và ghép lên trên khối lệnh.



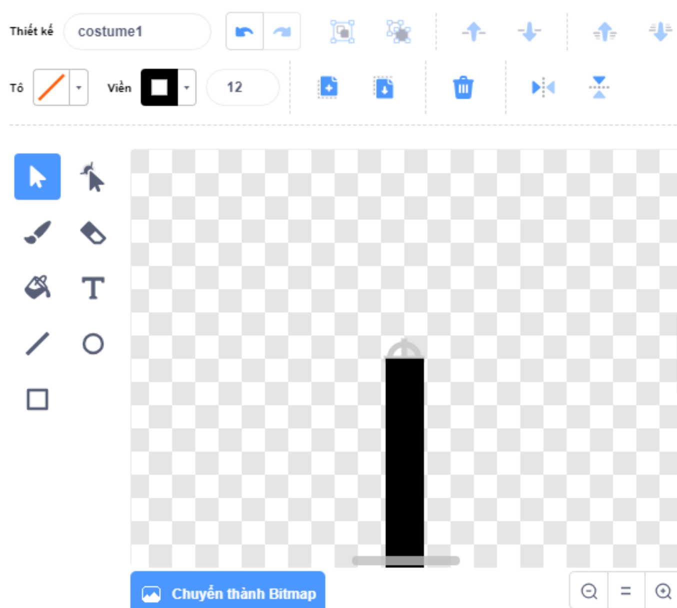
Hình 2.4.8. Đoạn khối lệnh hoàn chỉnh

BÀI 5 VỊ TRÍ BẮT ĐẦU CỦA CÁC ĐỐI TƯỢNG

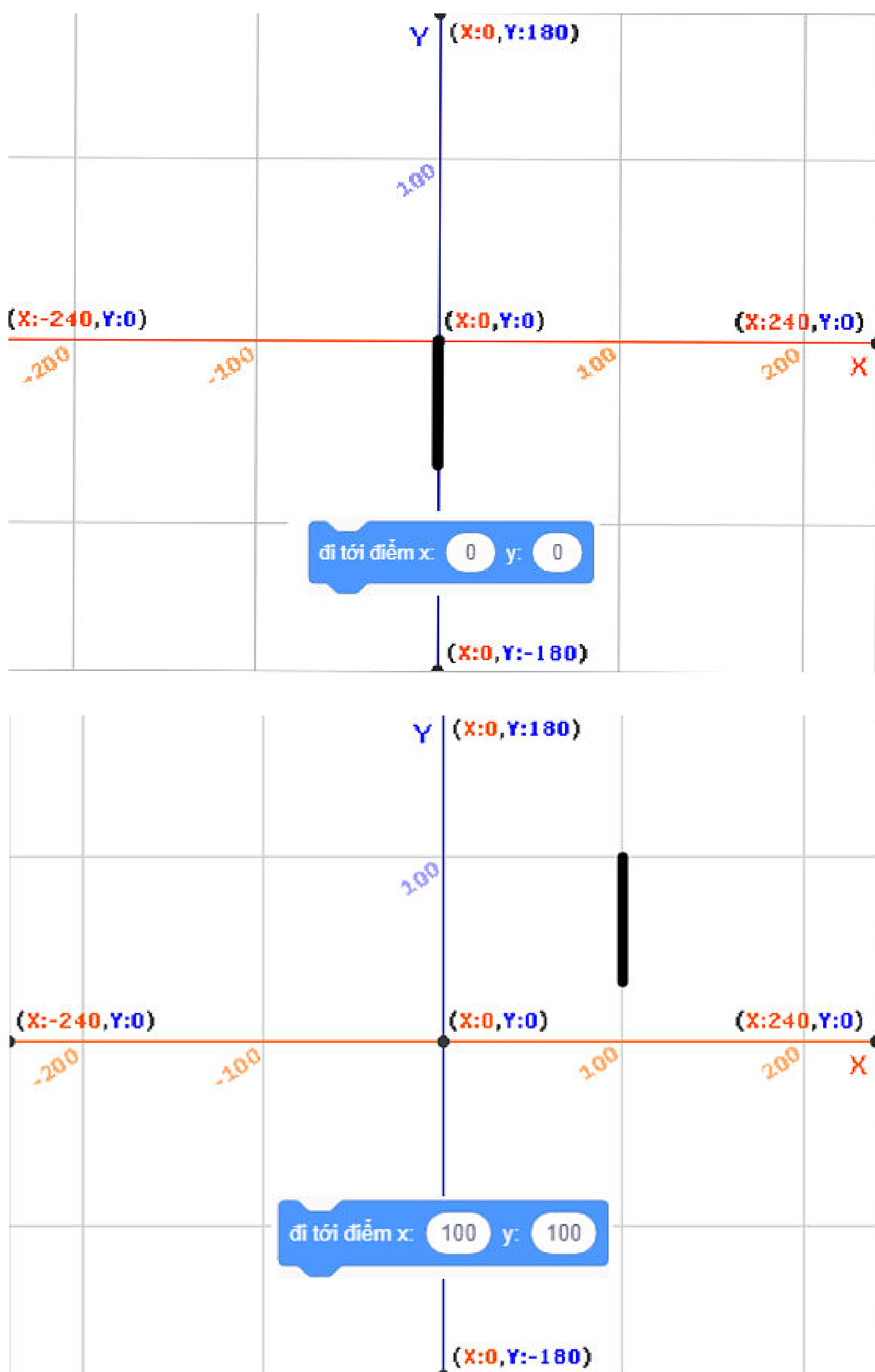
Trong rất nhiều trường hợp, chúng ta muốn khi bắt đầu chương trình (nháy vào nút lệnh) thì đối tượng sẽ tự động nhảy đến một vị trí cố định trên sân khấu. Bài học này hướng dẫn các bạn lập trình để thực hiện điều đó.

5.1. Nhảy tới vị trí cụ thể

Để lập trình cho nhân vật nhảy tới một vị trí cụ thể trên sân khấu, ta sử dụng khối lệnh **Đi tới điểm x, y** đi tới điểm x: 62 y: 170 trong nhóm **Chuyển động**. Khối lệnh này sẽ điều khiển nhân vật chuyển tới vị trí có toạ độ x, y trong khối lệnh ngay lập tức. Toạ độ của nhân vật chính là vị trí toạ độ của tâm nhân vật.

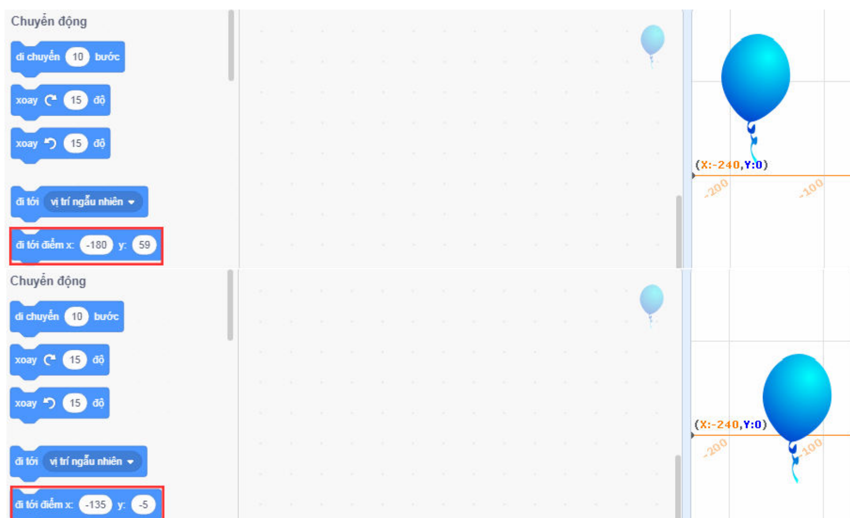


điểm đi tới điểm x: y: trong **Bảng danh mục các khối lệnh** sẽ tự động thay đổi tương ứng theo vị trí được kéo đến.



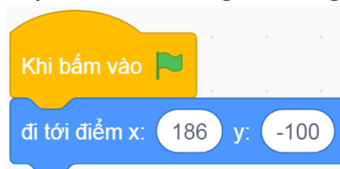
*Hình 2.5.1. Khối lệnh **Đi tới điểm***

Khi nhân vật được kéo đến một vị trí bất kì trên sân khấu, giá trị toạ độ x, y trong khối lệnh **Đi tới điểm** đi tới điểm x: y: trong **Bảng danh mục các khối lệnh** sẽ tự động thay đổi tương ứng theo vị trí được kéo đến.



*Hình 2.5.2: Khối lệnh **Đi tới điểm** tự động thay đổi giá trị x, y*

Như vậy, khi ta kéo nhân vật đến vị trí mong muốn thì khối lệnh [đi tới điểm x: ( ) y: ( )] sẽ thay đổi giá trị x, y thành toạ độ của vị trí mong muốn đó. Kéo khối lệnh sang khu lập trình, thêm khối lệnh [Khi bấm vào 🏳] lên bên trên khối lệnh để khi bắt đầu chương trình (nháy vào nút lệnh 🏳) nhân vật sẽ nhảy đến vị trí mong muốn ngay lập tức



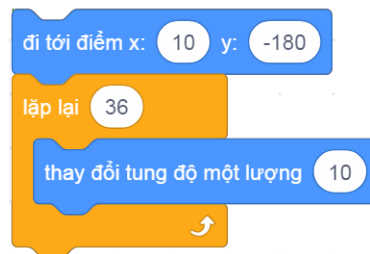
Hình 2.5.3. Đoạn khối lệnh hoàn chỉnh

BÀI 6 BÓNG BAY BAY LÊN TỪ VỊ TRÍ NGẪU NHIÊN

Sau khi học Bài 4, các bạn đã biết cách lập trình để quả bóng có thể bay lên, tuy nhiên sau khi bay lên quả bóng sẽ đứng yên ở đó. Bài học này sẽ hướng dẫn các bạn lập trình để điều khiển quả bóng lặp đi lặp lại việc bay lên và mỗi lần bay lên quả bóng lại xuất phát từ một vị trí khác.

6.1. Vòng lặp lồng nhau

Đoạn khối lệnh điều khiển quả bóng bay lên gồm hai thành phần: khối lệnh đi tới điểm đi tới điểm x: y: điều khiển vị trí quả bóng và vòng lặp thay đổi tung độ điều khiển quả bóng bay lên.



Hình 2.6.1. Điều khiển quả bóng bay lên

Chúng ta sử dụng khối lệnh **Liên tục** để lập trình việc nhảy tới vị trí mới và điều khiển quả bóng bay lên lặp đi lặp lại.

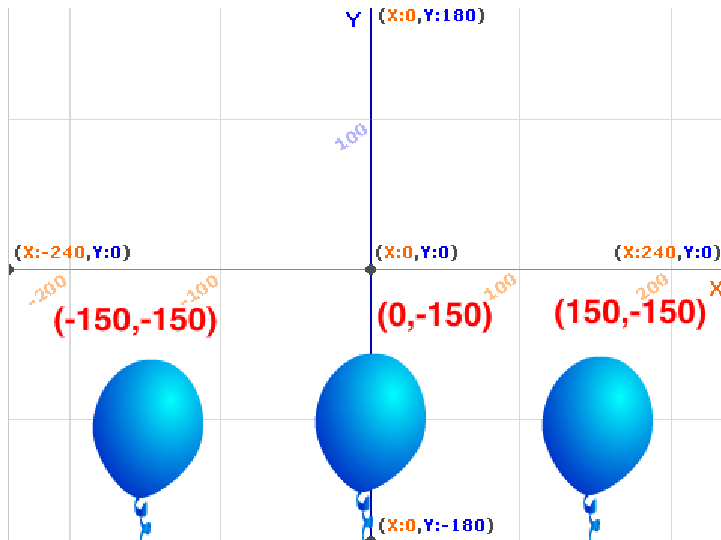


Hình 2.6.2. Đoạn khối lệnh lặp lồng nhau

Việc nhảy đến vị trí và bay lên được lặp đi lặp lại mãi mãi cho đến khi ta nháy vào nút lệnh **Dừng** .

6.2. Lấy giá trị ngẫu nhiên

Khi vị trí x thay đổi, vị trí theo phương ngang của nhân vật sẽ bị thay đổi.



Hình 2.6.3. Thay đổi vị trí x

Chính vì vậy, để thay đổi vị trí bay lên của quả bóng, các bạn cần thay đổi hoành độ (vị trí x) sau mỗi lần lặp (mỗi lần bay lên). Chúng ta thay đổi hoành độ một cách ngẫu nhiên bằng khối lệnh **Lấy ngẫu nhiên** [lấy ngẫu nhiên từ ( ) đến ( )]. Khối lệnh này sẽ chứa một giá trị được lấy ngẫu nhiên trong khoảng từ số thứ nhất đến số thứ hai. Vị trí theo chiều ngang (trục x) của đối tượng Bóng bay bị giới hạn trong khu vực sân khấu với x có giá trị từ –240 đến 240. Chúng ta ghép khối lệnh đã được chỉnh sửa giá trị [lấy ngẫu nhiên từ (-240) đến (240)] vào giá trị x của khối lệnh [đi tới điểm x: ( ) y: (-180)].

Vậy là sau mỗi lần lặp, khối lệnh đi tới [đi tới điểm x: (lấy ngẫu nhiên từ (-240) đến (240)) y: (-180)] sẽ không còn khiến nhân vật nhảy đến một vị trí cố định nữa mà sẽ có vị trí theo chiều ngang khác nhau.



Hình 2.6.4. Đoạn khối lệnh hoàn chỉnh

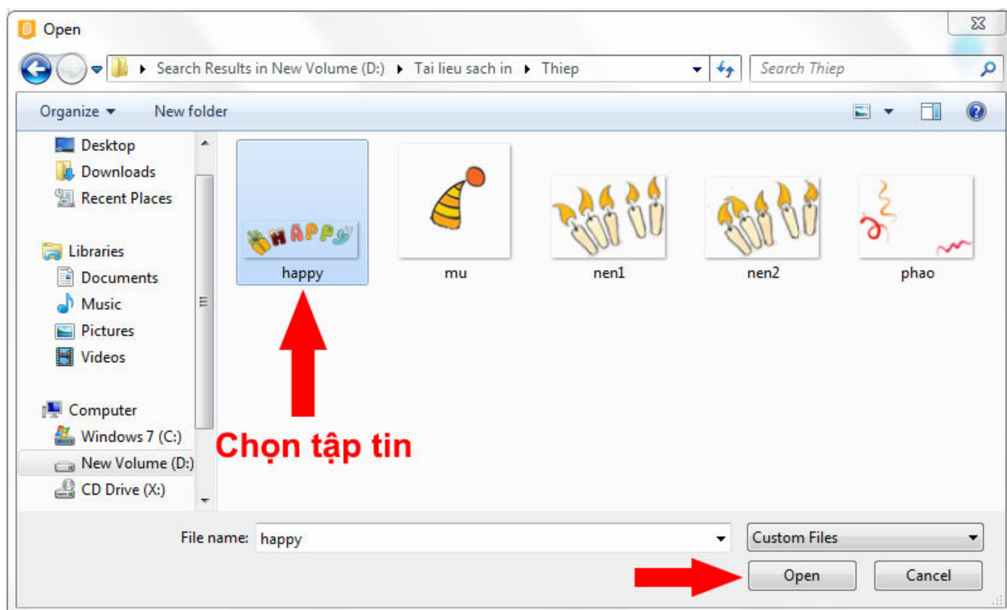
BÀI 7 LẬP TRÌNH ĐỐI TƯỢNG NẾN

Trong bài học này, các bạn sẽ được hướng dẫn cách thêm nhân vật chữ Happy và nhân vật Nến từ thư mục trên máy tính, đồng thời lập trình nến có thể cháy.

7.1. Thêm nhân vật từ thư mục

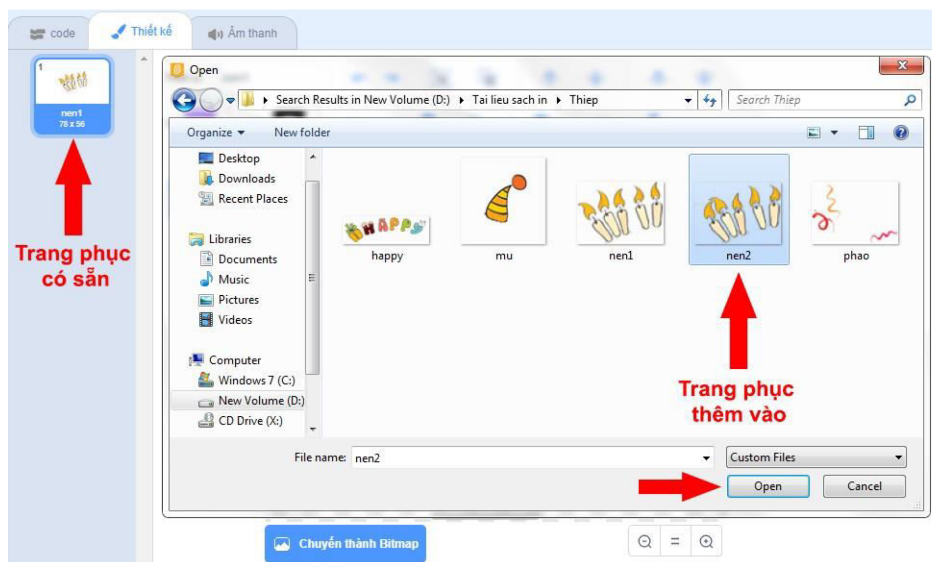
Đầu tiên, ta truy cập website để tải tài liệu về máy tính.

Ở phần danh sách nhân vật, ta nháy vào nút , cửa sổ mới sẽ hiện lên để chọn tập tin. Ta tìm đến thư mục tài liệu vừa tải về, chọn tập tin rồi nháy Open.



Hình 2.7.1. Thêm đối tượng chữ Happy từ thư mục

Tiếp theo là thêm ngọn nến. Nến có hai trang phục, vậy nên sau khi thêm nhân vật nến thì ta cần thêm một trang phục nữa cho Nến. Ở khu thiết kế của đối tượng Nến ta nháy vào nút và chọn trang phục thứ hai của Nến.



Hình 2.7.2. Thêm trang phục cho nến

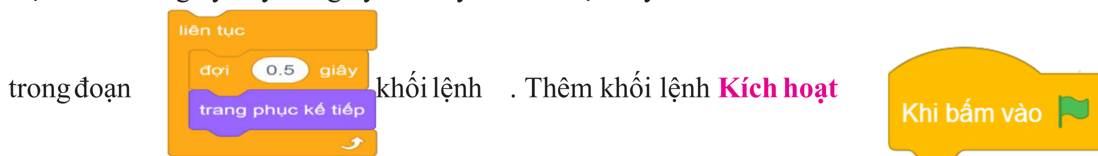
7.2. Lập trình cho đối tượng Nến

Ngọn nến được lập trình chuyển đổi giữa hai trang phục liên tục, khi đó trên sân khấu ngọn nến sẽ như đang bập bùng cháy. Để thay đổi trang phục, chúng ta sử dụng khối lệnh **Trang phục kế tiếp** trang phục kế tiếp trong nhóm **Hiển thị**. Khi nháy chạy khối lệnh, nhân vật Nến sẽ thay đổi hình ảnh của mình. Khi khối lệnh chạy liên tục ta sẽ thấy ngọn nến như đang cháy.



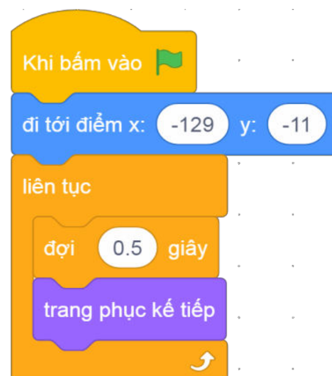
Hình 2.7.3. Hai trang phục của đối tượng Nến

Chúng ta sử dụng khối lệnh **Liên tục** để ngọn nến luôn cháy, tuy nhiên khi chạy thử thì ngọn nến thay đổi trang phục rất nhanh và không đẹp. Các bạn ghép thêm khối lệnh đợi đợi 1 giây vào trong vòng lặp để ngọn nến cháy chậm và đẹp hơn. Chúng tacó thể chỉnh sửa thời gian đợi thành 0.5 giây hay 0.2 giây để thay đổi tốc độ cháy như



trong đoạn khối lệnh . Thêm khối lệnh **Kích hoạt**

và **Đi tới điểm** để khi bắt đầu chương trình ngọn nến sẽ nhảy đến vị trí mong muốn.



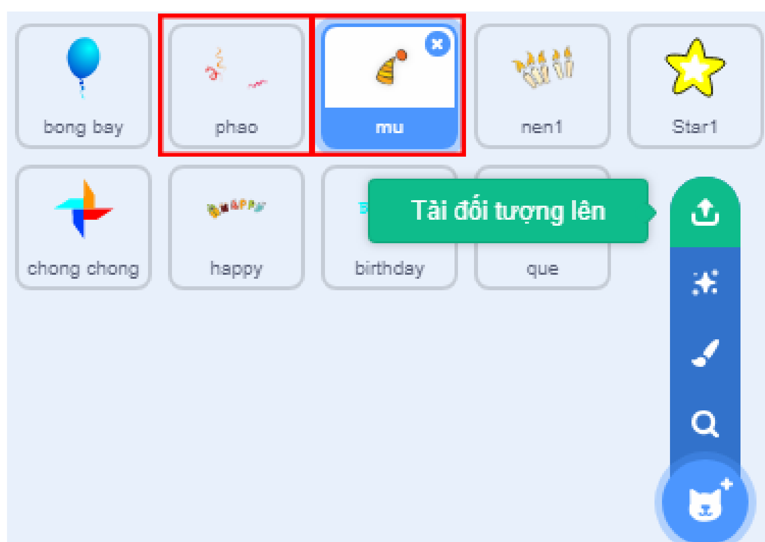
Hình 2.7.4. Đoạn khối lệnh hoàn chỉnh

BÀI 8 ĐỐI TƯỢNG MŨ VÀ PHÁO

Bài học này sẽ hướng dẫn các bạn lập trình hai đối tượng Mũ và Pháo. Hai nhân vật này đều có chuyển động lên và xuống lặp đi lặp lại khá giống nhau.

8.1. Thêm nhân vật từ thư mục

Cách thêm nhân vật Mũ và Pháo giống với nhân vật Nến.



Hình 2.8.1. Thêm nhân vật Mũ và nhân vật Pháo

8.2. Lập trình cho Mũ

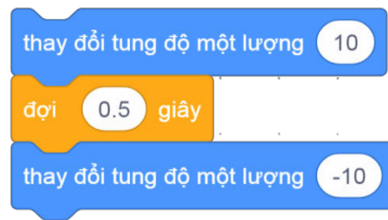
Chiếc mũ sẽ chuyển động lên và xuống đều đặn. Qua bài 4, chúng ta đã biết muốn điều khiển đối tượng di chuyển theo phương thẳng đứng sẽ phải thay đổi tung độ (vị trí y) của nhân vật, vậy ta cần sử dụng hai khối lệnh thay đổi tung độ

thay đổi tung độ một lượng ( ) . Trong hai khối lệnh này, một khối lệnh sẽ làm tăng giá trị tung độ (lượng giá trị tung độ thay đổi là số dương) để nhân vật chuyển động theo hướng từ dưới lên trên và khối lệnh còn lại sẽ giảm giá trị tung độ (lượng giá trị tung độ thay đổi là số âm) để nhân vật chuyển động theo hướng từ trên xuống dưới.

thay đổi tung độ một lượng (10)
thay đổi tung độ một lượng (-10)

Hình 2.8.2. Đoạn khối lệnh di chuyển của Mũ

Khi chạy đoạn khối lệnh này, ta sẽ thấy nhân vật vẫn đứng im, vì các khối lệnh thực hiện rất nhanh, vừa di chuyển lên với khối lệnh thay đổi tung độ một lượng 10 nhân vật đã di chuyển xuống bằng khối lệnh thay đổi tung độ một lượng -10 ngay lập tức. Chính vì vậy, chúng ta ghép thêm khối lệnh đợi đợi 1 giây vào giữa hai khối lệnh trên để chuyển động được quan sát rõ ràng hơn. Ta có thể tuỳ chỉnh thời gian đợi để thay đổi tốc độ lên và xuống của chiếc mũ.



Hình 2.8.3. Thêm khối lệnh đợi giữa hai lần di chuyển

Để việc di chuyển lên và xuống này được thực hiện liên tục, ta ghép thêm vòng lặp bao quanh bên ngoài đoạn khối lệnh. Hoạt động của đoạn khối lệnh có thêm vòng lặp được mô tả ở hình 2.8.2.



Hình 2.8.4. Hoạt động của vòng lặp

Giữa các lần lặp với nhau không có khối lệnh đợi, sau khi khối lệnh thay đổi tung độ một lượng -10 được chạy, ngay lập tức khối lệnh thay đổi tung độ một lượng 10 cũng được thực hiện, chúng ta không kịp nhìn thấy chiếc mũ di chuyển xuống. Vậy nên chúng ta ghép thêm một khối lệnh đợi 0.5 giây nữa.



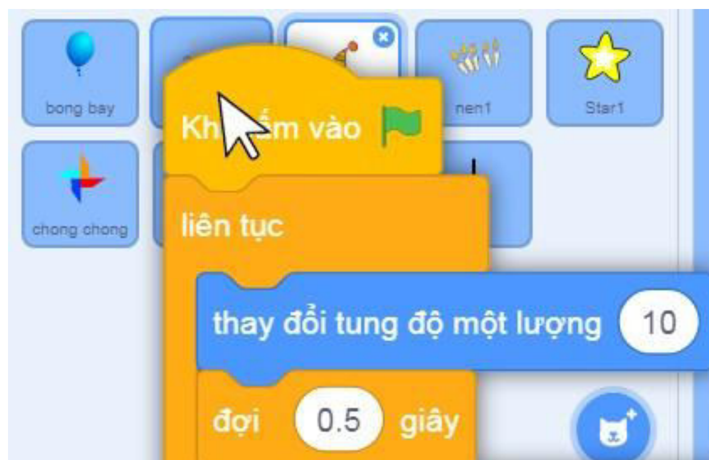
*Hình 2.8.5. Thêm khối lệnh **Đợi***

Vậy là đoạn khối lệnh cho việc di chuyển đã xong, ta cần thêm khối lệnh **Kích hoạt** Khi bấm vào  cho đoạn khối lệnh của mình.

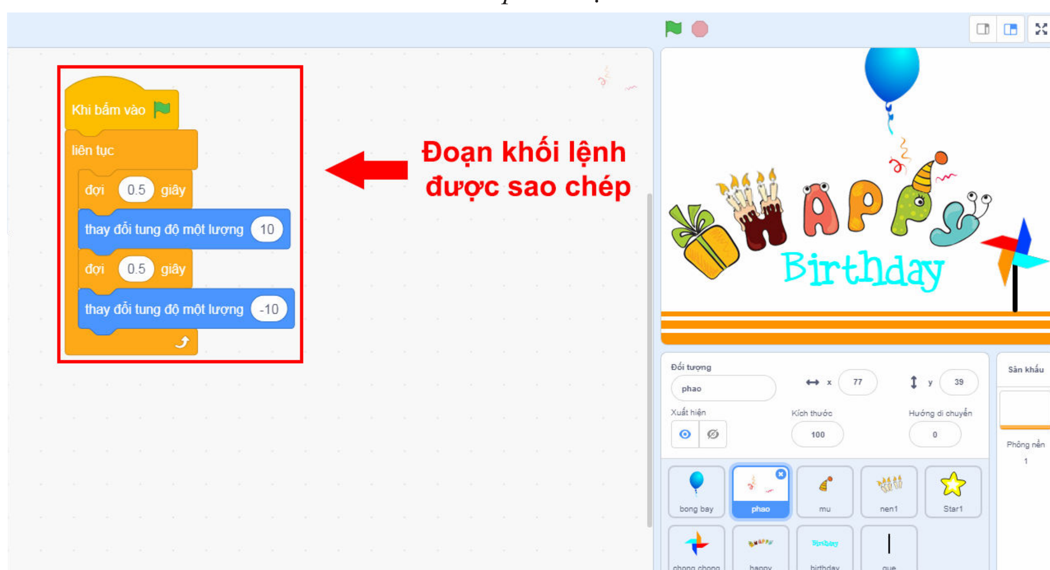
8.3. Lập trình cho Pháo

Pháo được lập trình di chuyển lên và xuống liên tục giống chiếc mũ, khi Mũ di chuyển lên thì Pháo di chuyển xuống và ngược lại. Ta sao chép đoạn khối lệnh của Mũ, sau đó kéo sang nhân vật Pháo bằng cách nháy nút phải chuột lên khối lệnh trên cùng Khi bấm vào , chọn **Nhân bản**. Khối lệnh mới xuất hiện giống khối lệnh cũ được gắn vào con trỏ chuột mà không cần phải nhấn giữ chuột. Di chuyển con trỏ chuột và khối lệnh đến ảnh thu nhỏ của nhân vật Pháo và nháy chuột vào đó, khối lệnh sẽ được xuất hiện bên khu lập trình của nhân vật Pháo.

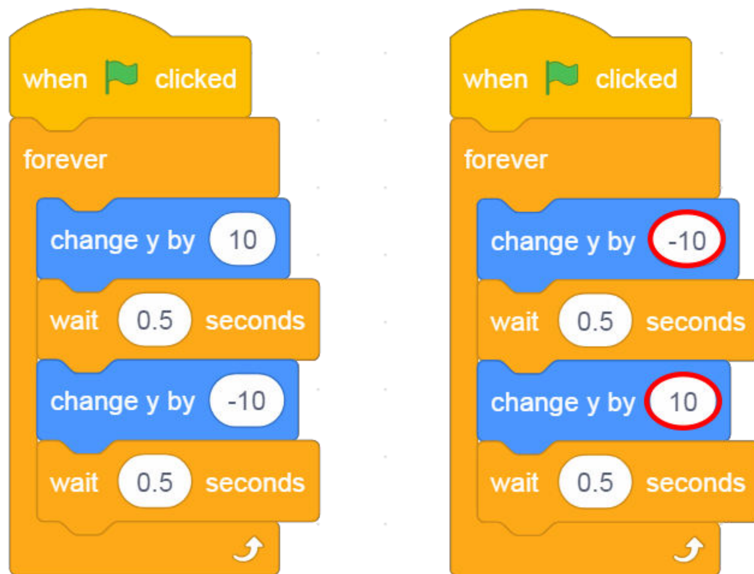




Hình 2.8.6. Sao chép khối lệnh



Hình 2.8.7. Kết quả



Khi chạy chương trình, Mũ và Pháo trên sân khấu có chuyển động giống hệt nhau vì chúng có đoạn khối lệnh điều khiển giống nhau. Để chuyển động của Pháo ngược với Mũ, các bạn đổi chỗ hai giá trị của hai khối lệnh thay đổi tung độ thay đổi tung độ một lượng trong đoạn khối lệnh của Pháo. Đối tượng Mũ có khối lệnh di chuyển đầu tiên là di chuyển lên, đối tượng Pháo có khối lệnh di chuyển đầu tiên là di chuyển xuống.

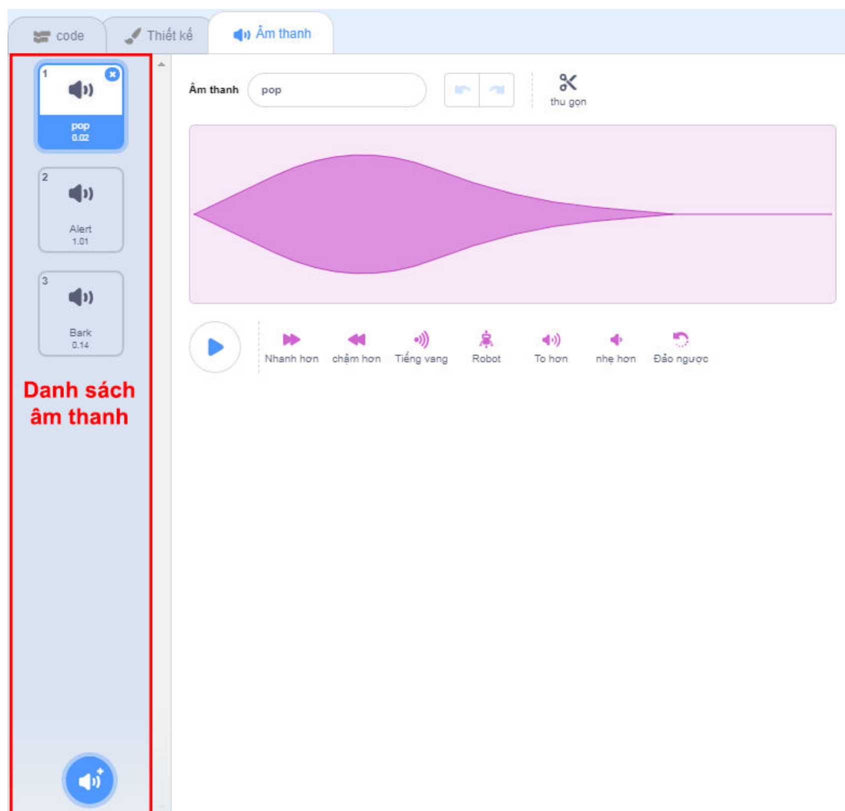
Hình 2.8.8. Đoạn khối lệnh hoàn chỉnh của Mũ và Pháo

BÀI 9 ÂM THANH

Bài học này sẽ hướng dẫn các bạn lập trình để thêm âm thanh cho đối tượng và thực hành để thêm nhạc vào thiệp.

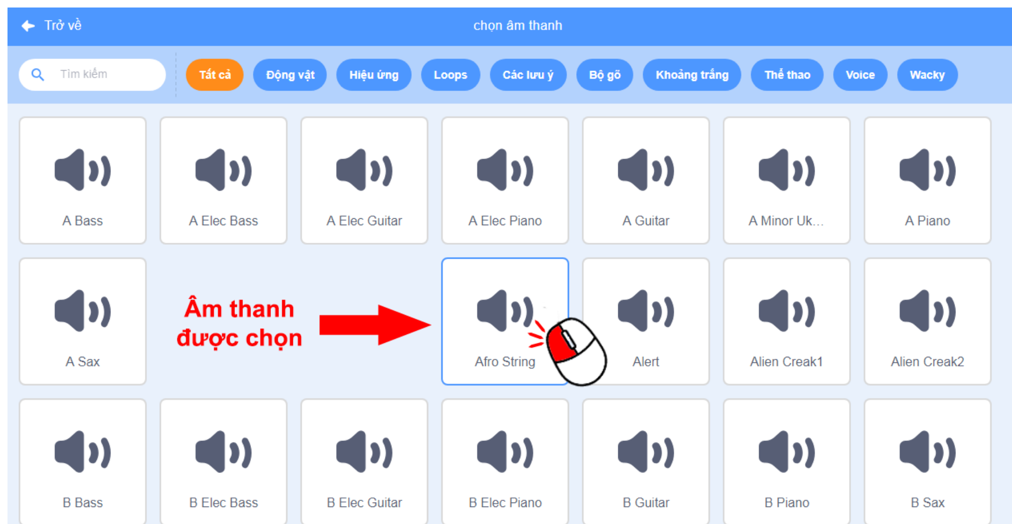
9.1. Thêm âm thanh

Mỗi nhân vật, đối tượng đều có danh sách âm thanh riêng ở **Khu âm thanh** (trang **Sounds**). Đây là nơi chứa tất cả âm thanh mà nhân vật có thể chơi. Mỗi nhân vật được thêm mới (từ thư viện, tạo mới, tải từ máy tính hay chụp ảnh) đều được tạo sẵn âm thanh ngầm định.



Hình 2.9.1. Danh sách âm thanh ở ***Khu âm thanh***

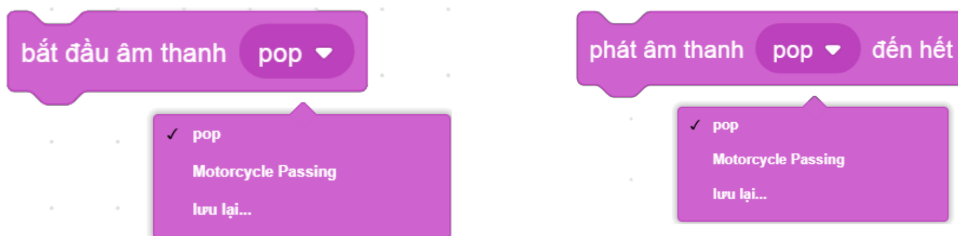
Âm thanh của nhân vật nào sẽ được thêm vào danh sách âm thanh của nhân vật đó, còn nhạc nền thì sẽ đưa vào danh sách âm thanh của sân khấu. Chương trình Thiệp sinh nhật có nhạc nền là bài Happy Birthday là âm thanh đã có sẵn trong thư viện âm thanh, chúng ta sẽ thêm âm thanh có sẵn trong thư viện bằng cách nháy vào nút .



Hình 2.9.2. Thêm nhạc birthday cho thiệp

9.2. Lập trình

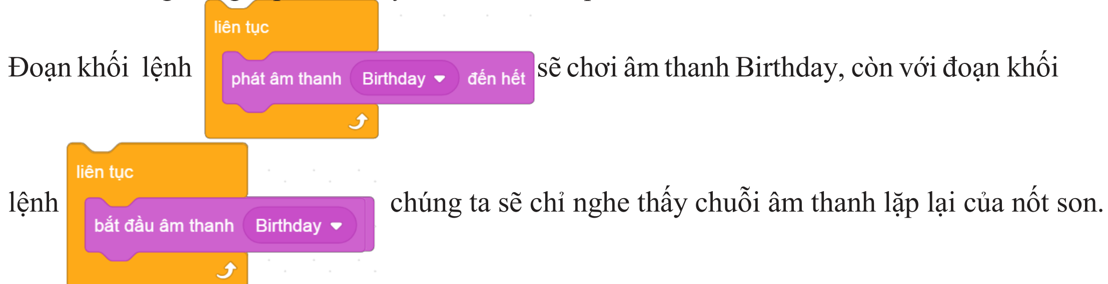
Trong Danh mục các nhóm lệnh, nhóm **Âm thanh** có hai khối lệnh dùng để chơi âm thanh là bắt đầu âm thanh pop và phát âm thanh pop đến hết. Cả hai khối lệnh đều dùng để chơi âm thanh có trong danh sách âm thanh của đối tượng sân khấu.



Hình 2.9.3. Hai khối lệnh chơi âm thanh

Âm thanh "birthday" chơi liên tục cho đến khi dừng chương trình, vì vậy ta ghép khối lệnh liên tục bao bên ngoài khối lệnh chơi âm thanh. Hai khối lệnh chơi âm thanh được ghép vào trong vòng lặp, khi chạy sẽ có hai kết quả khác nhau.

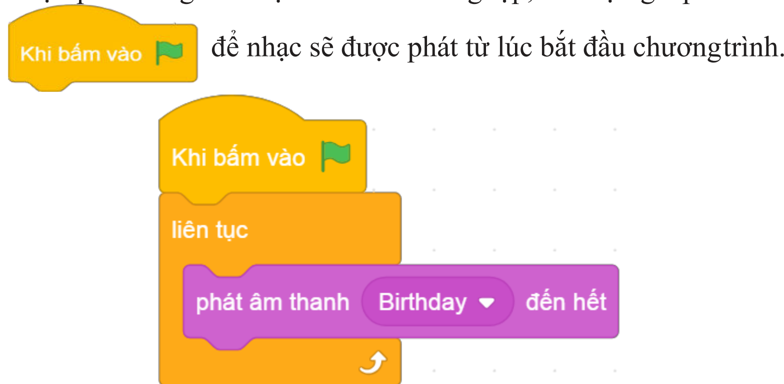
Đoạn khối lệnh sẽ chơi âm thanh Birthday, còn với đoạn khối lệnh chúng ta sẽ chỉ nghe thấy chuỗi âm thanh lặp lại của nốt son.



Khi một âm thanh đang được chơi, nếu chạy lại khối lệnh bắt đầu âm thanh đó thì âm thanh

sẽ được chơi lại từ đầu. Khối lệnh phát âm thanh Birthday đến hết sẽ chơi âm thanh được chọn và đợi cho đến khi âm thanh đó được chơi hết mới thực hiện công việc các khối lệnh tiếp theo (trong trường hợp này là lặp lại việc chơi lại âm thanh birthday). Còn khối bắt đầu âm thanh Birthday sẽ chơi âm thanh và thực hiện các công việc tiếp theo luôn mà không chờ cho đến khi âm thanh được chơi hết. Ở trong vòng lặp, khối lệnh này sẽ liên tục được thực hiện, âm thanh liên tục được chơi lại từ đầu nên chúng ta sẽ chỉ nghe thấy chuỗi âm thanh lặp lại của nốt son.

Sau khi biết được phải dùng khối lệnh nào cho vòng lặp, các bạn ghép thêm khối lệnh kích hoạt Khi bấm vào để nhạc sẽ được phát từ lúc bắt đầu chươngtrình.



Hình 2.9.4. Đoạn khối lệnh hoàn chỉnh

BÀI 10 LẬP TRÌNH VỚI BẢN SAO

Đối tượng cuối cùng của tấm thiệp là Ngôi sao. Bài học này sẽ hướng dẫn các bạn lập trình điều khiển Ngôi sao, tạo bản sao nhấp nháy của Ngôi sao.

10.1. Thêm đối tượng Ngôi sao

Trong thư viện đã có sẵn Ngôi sao, các bạn thêm nhân vật từ thư viện, chọn mục Holidays sẽ thấy Ngôi sao cần tìm.



Hình 2.10.1. Thêm Ngôi sao từ thư viện

Sau khi thêm, các bạn có thể phóng to hoặc thu nhỏ ngôi sao để chỉnh kích thước cho phù hợp.

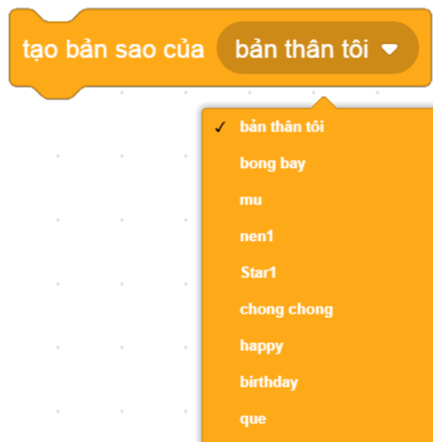
10.2. Tạo bản sao

Để tấm thiệp có nhiều ngôi sao, chúng ta có thể thêm nhiều nhân vật ngôi sao khác cũng giống như vậy, tuy nhiên việc đó mất rất nhiều công sức và dễ gây nhầm lẫn các nhân vật. Thay vào đó, chúng ta sẽ sử dụng khối lệnh **Tạo bản sao** [tạo bản sao của bản thân tôi ▾] để tạo ra nhiều ngôi sao khác giống hệt ngôi sao ban đầu. Ở khu lập trình của Ngôi sao, các bạn chạy thử khối lệnh **Tạo bản sao**. Ta kéo ngôi sao ra sẽ thấy một ngôi sao khác xuất hiện ngay bên dưới.



Hình 2.10.2. Bản sao của Ngôi sao xuất hiện bên dưới bản chính

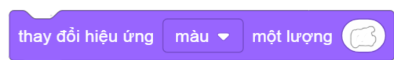
Ta có thể chọn nhân vật trong danh sách để tạo bản sao của nhân vật khác. Tất cả các bản sao đều giống hệt bản chính và được đặt ra sau bản chính một lớp.



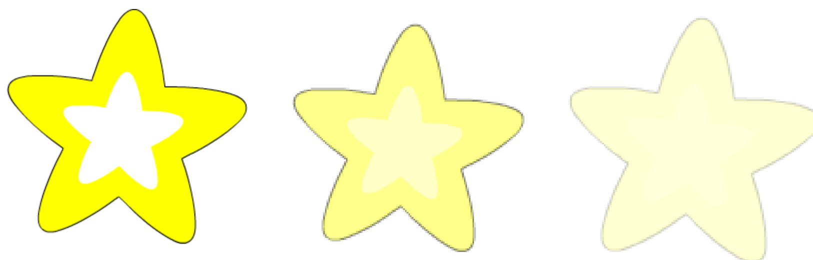
Hình 2.10.3. Tạo bản sao từ nhân vật khác

10.3. Lập trình thay đổi hiệu ứng cho bản sao

Chúng ta sẽ lập trình điều khiển ngôi sao mờ đi, một giây sau sẽ hiện rõ trở lại và một giây sau nữa sẽ đổi màu. Chúng ta sẽ sử dụng khối lệnh Thay đổi hiệu ứng



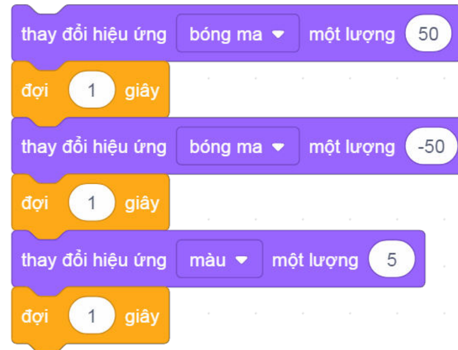
điều khiển việc thay đổi hiệu ứng màu và hiệu ứng bóng ma. Với hiệu ứng bóng ma, lượng giá trị thay đổi là số dương sẽ khiến đối tượng trở nên mờ hơn và ngược lại. Ta có thể chọn lượng giá trị thay đổi của hiệu ứng tuỳ ý để ngôi sao đẹp hơn.



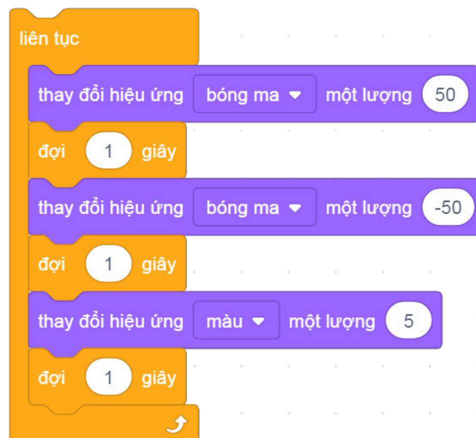
Hình 2.10.4. Các mức độ hiệu ứng mờ ảo

Chúng ta sẽ cho ngôi sao nhấp nháy bằng hai khối lệnh thay đổi hiệu ứng bóng ma và đổi màu bằng một khối lệnh đổi hiệu ứng màu. Ta ghép các khối lệnh và chạy thử.

Hình 2.10.5. Thay đổi các hiệu ứng



Ghép thêm vòng lặp để việc thay đổi hiệu ứng được thực hiện liên tục.



Hình 2.10.6. Thay đổi các hiệu ứng liên tục

Tuy nhiên, đoạn khối lệnh trên chỉ giúp Ngôi sao nhấp nháy mà bản sao của nó thì không. Ghép khối lệnh khi tôi bắt đầu là một bản sao lên trên để bản sao của Ngôi sao nhấp nháy được như vậy.



Hình 2.10.7. Lập trình cho bản sao

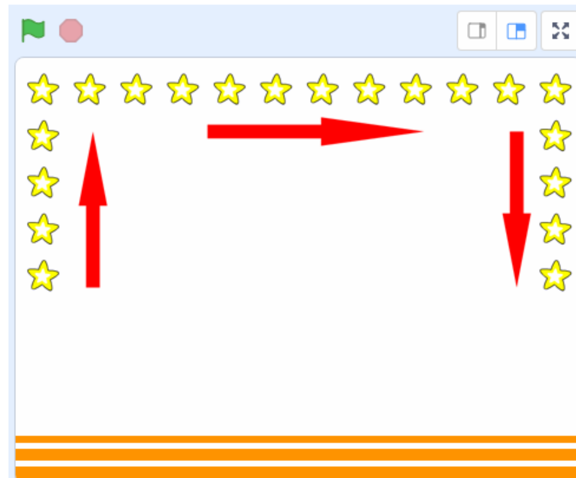
Lúc này khi các bạn tạo bản sao cho Ngôi sao, kéo Ngôi sao ra sẽ thấy bản sao của Ngôi sao thay đổi hiệu ứng liên tục.



Hình 2.10.8. Bản sao thay đổi hiệu ứng

10.4. Sắp xếp các bản sao của Ngôi sao

Trong tấm thiệp, các ngôi sao được sắp xếp thành hai cột và một hàng giống như ba cạnh của hình chữ nhật. Ta lập trình cho ngôi sao xếp lần lượt từng cạnh một bằng cách lặp lại công việc tạo bản sao và di chuyển sang vị trí tiếp theo.



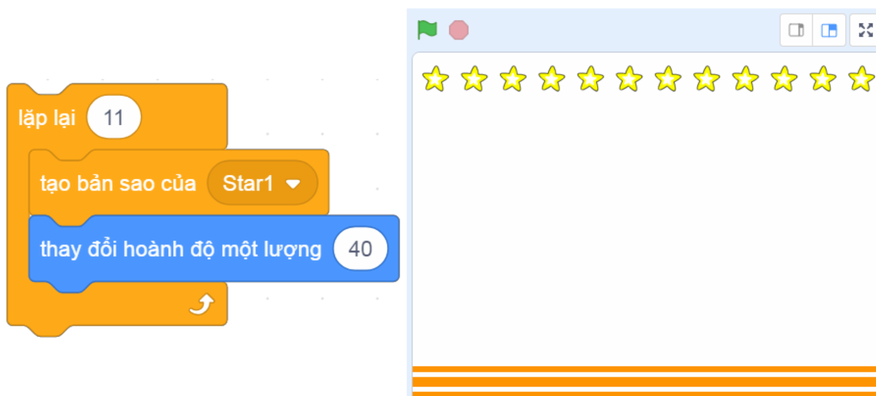
Hình 2.10.9. Sắp xếp các bản sao

Với cột đầu tiên, Ngôi sao được lập trình di chuyển từ dưới lên trên, tức là giá trị *tung độ* của Ngôi sao được tăng.



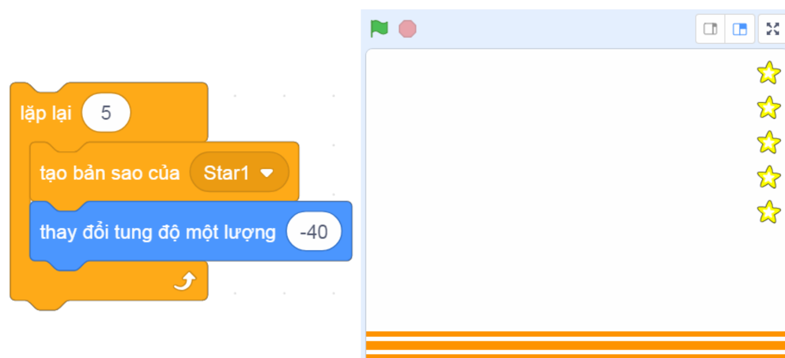
Hình 2.10.10. Cột sao thứ nhất

Hàng tiếp theo, Ngôi sao di chuyển từ trái sang phải, tức là giá trị *hoành độ* của ngôi sao được tăng.



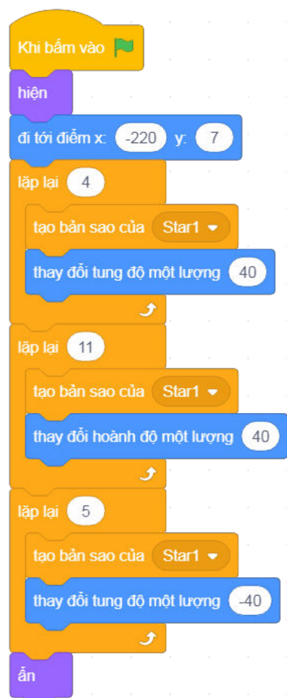
Hình 2.10.11. Hàng sao

Cột cuối cùng, Ngôi sao di chuyển từ trên xuống dưới, tức là giảm giá trị tung độ của Ngôi sao.



Hình 2.10.12. Cột sao thứ 2

Chỉ bản sao mới có hiệu ứng nên sau khi tạo đủ bản sao rồi, các bạn ẩn ngôi sao đi bằng khối lệnh **Ẩn** trong nhóm lệnh **Ngoại hình**. Tuy nhiên để tránh việc khi chạy lại chương trình lần sau Ngôi sao vẫn bị ẩn thì ta thêm khối lệnh **Hiện** vào đầu đoạn khối lệnh. Cuối cùng là khối lệnh kích hoạt và đặt vị trí bắt đầu cho Ngôi sao.



Hình 2.10.13. Đoạn khối lệnh hoàn chỉnh

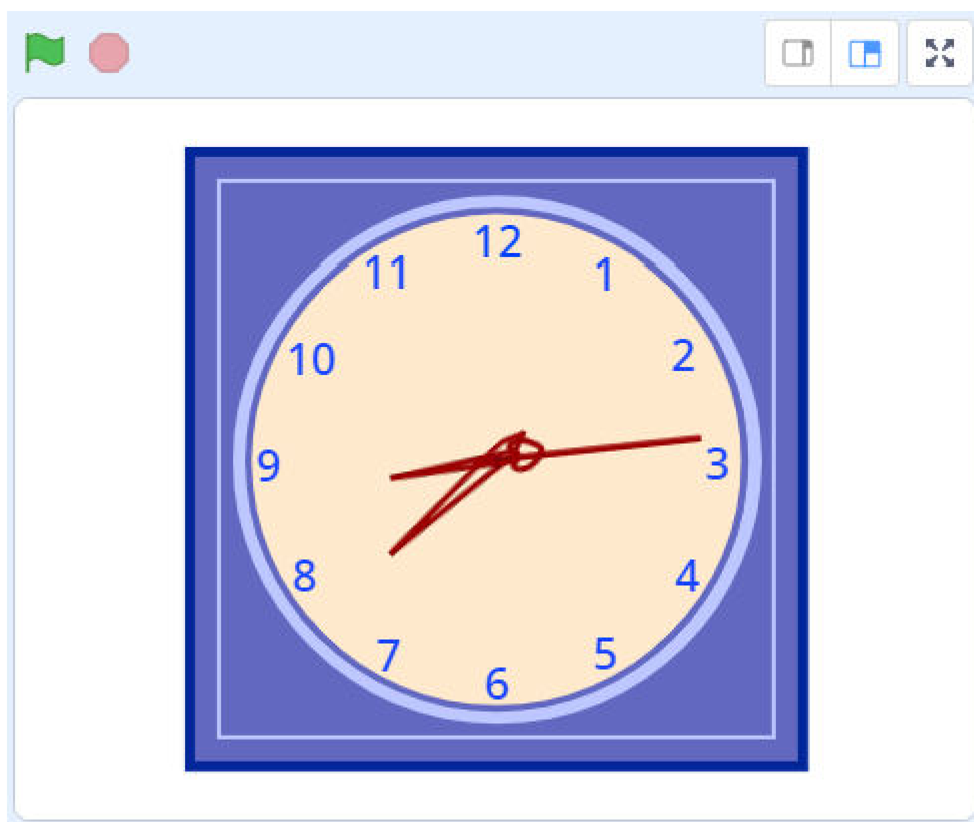
Chương thiết kế **Thiệp sinh nhật** đến đây là kết thúc, chúng ta đã được trang bị những kiến thức cơ bản để có thể tự thiết kế một tấm thiệp cho riêng mình.

CHƯƠNG III

Đồng hồ

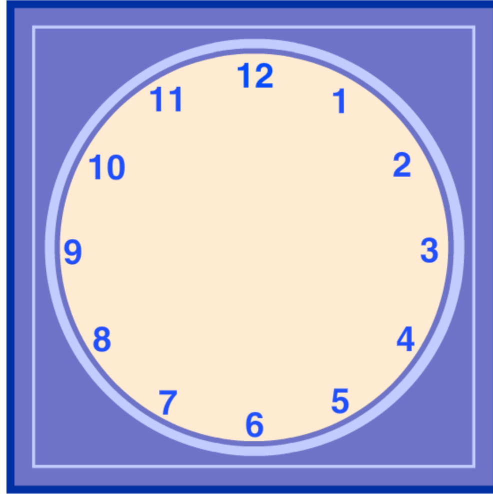
Trong chương này, các bạn sẽ được hướng dẫn để tạo ra một chiếc đồng hồ kim. Sau đó, các bạn sẽ học được cách thiết kế các thành phần của đồng hồ và lập trình cho các kim để đồng hồ hoạt động với thời gian chính xác như thời gian được cài đặt trên máy tính.

Tổng quan chương trình



Chiếc đồng hồ gồm bốn đối tượng thành phần: Mặt đồng hồ, kim giờ, kim phút và kim giây.

1. Mặt đồng hồ



Hình dạng, màu sắc và các số trên đồng hồ được biểu diễn ở đối tượng Mặt đồng hồ.

2. Kim giờ



Đối tượng Kim giờ mô phỏng chuyển động của kim giờ.

3. Kim phút



Đối tượng Kim phút mô phỏng chuyển động của kim phút.

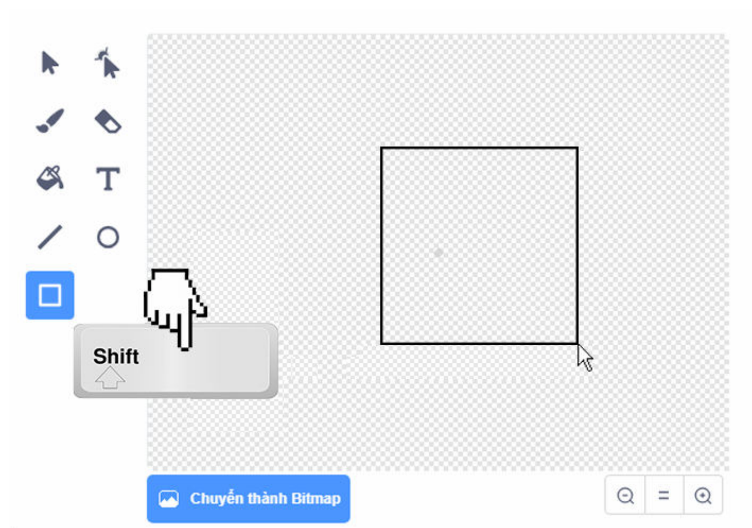
4. Kim giây



Đối tượng Kim giây mô phỏng chuyển động của kim giây.

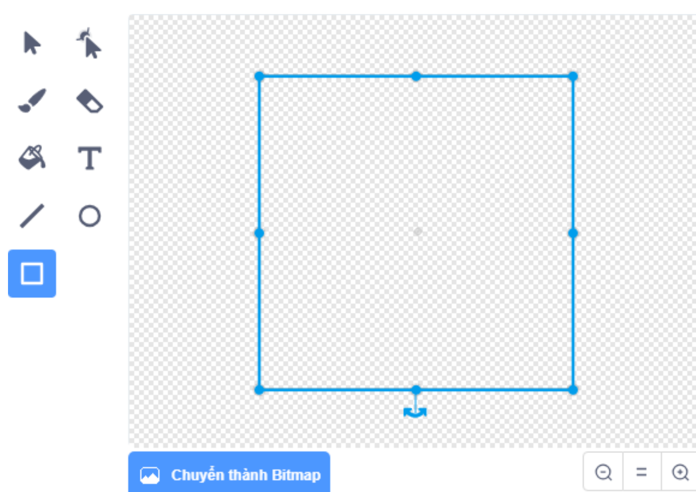
BÀI 1 VẼ MẶT ĐỒNG HỒ

Mặt đồng hồ được tạo bởi hai hình vuông và hai đường tròn bên trong. Trước tiên,chúng ta cần vẽ đường viền hình vuông ngoài cùng bằng cách chọn công cụ **Hình chữ nhật** ☐ , kéothả chuột đồng thời nhấn giữ phím **Shift** để vẽ được hình vuông.



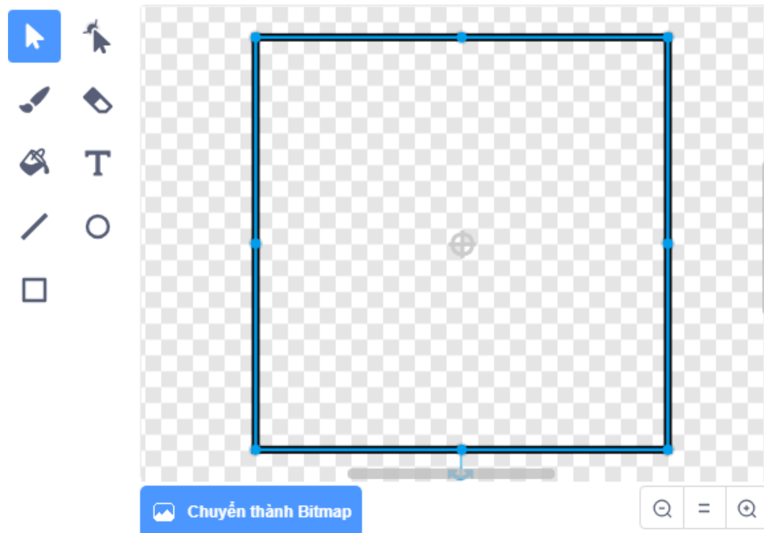
Hình 3.1.1. Vẽ hình vuông

Chỉnh lại tâm hình vuông trùng với điểm chấm tròn ở chính giữa.



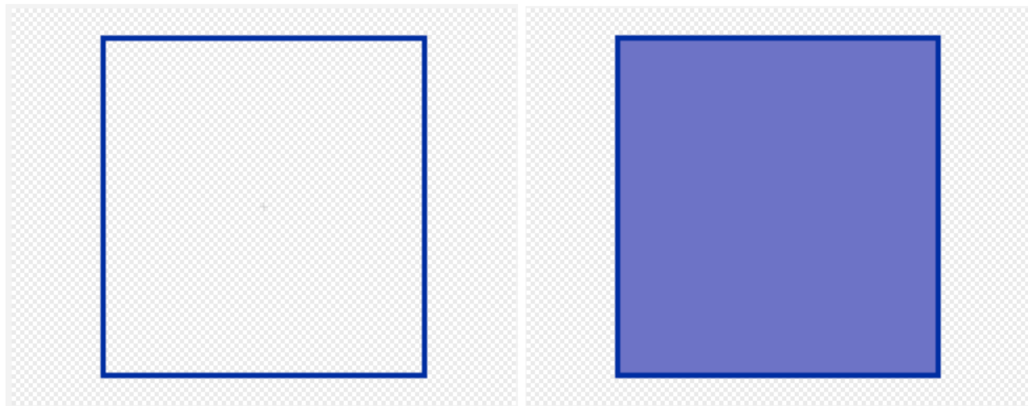
Hình 3.1.2. Chỉnh tâm hình vuông

Di chuyển hình vuông trên trang vẽ để điểm chính giữa hình vuông trùng với tâm trang vẽ. Các bạn có thể sử dụng chức năng phóng to [zoom-in icon] để điều chỉnh dễ dàng hơn.



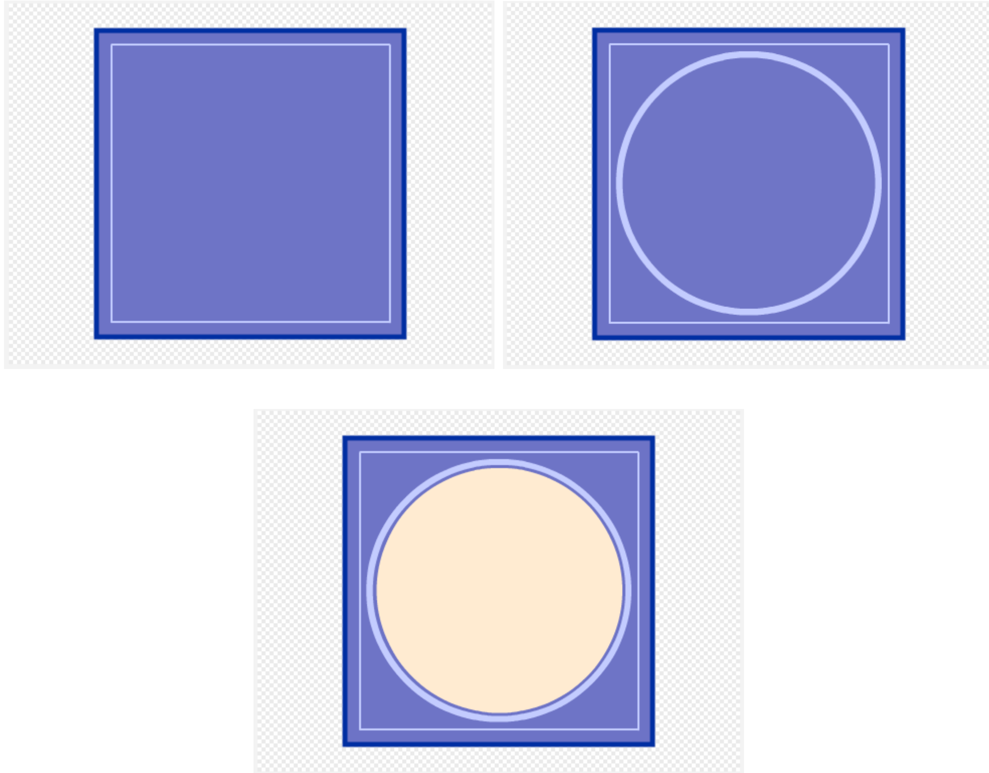
Hình 3.1.3. Chỉnh tâm hình vuông bằng tâm trang vẽ

Tô màu bên trong hình vuông bằng công cụ [fill-tool icon], chỉnh màu và kích thước nét vẽ đường viền tuỳ ý.



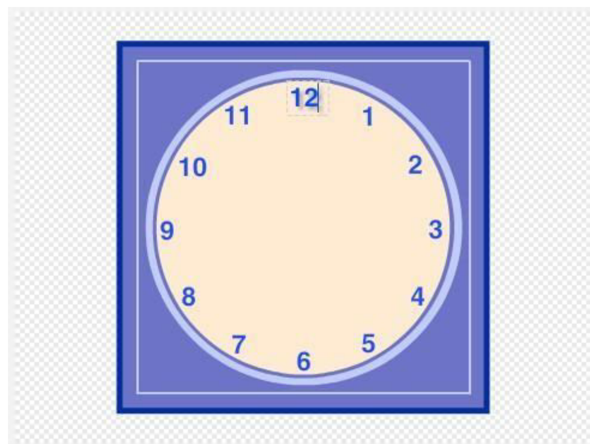
Hình 3.1.4. Chỉnh kích thước, màu đường viền và tô màu

Vẽ đường viền hình vuông thứ hai và thêm hai đường tròn. Vẽ hình tròn bằng công cụ vẽ hình elip [ellipse-tool icon], khi vẽ nhấn phím **Shift** để có được hình tròn. Thao tác chỉnh tâm và chỉnh màu được thực hiện tương tự.



Hình 3.1.5. Thêm một đường viền hình vuông và hai đường tròn

Thêm số cho đồng hồ. Dùng công cụ T để thêm các số từ 1 đến 12 và điều chỉnh vị trí của số cho phù hợp bằng các phím mũi tên trên bàn phím.



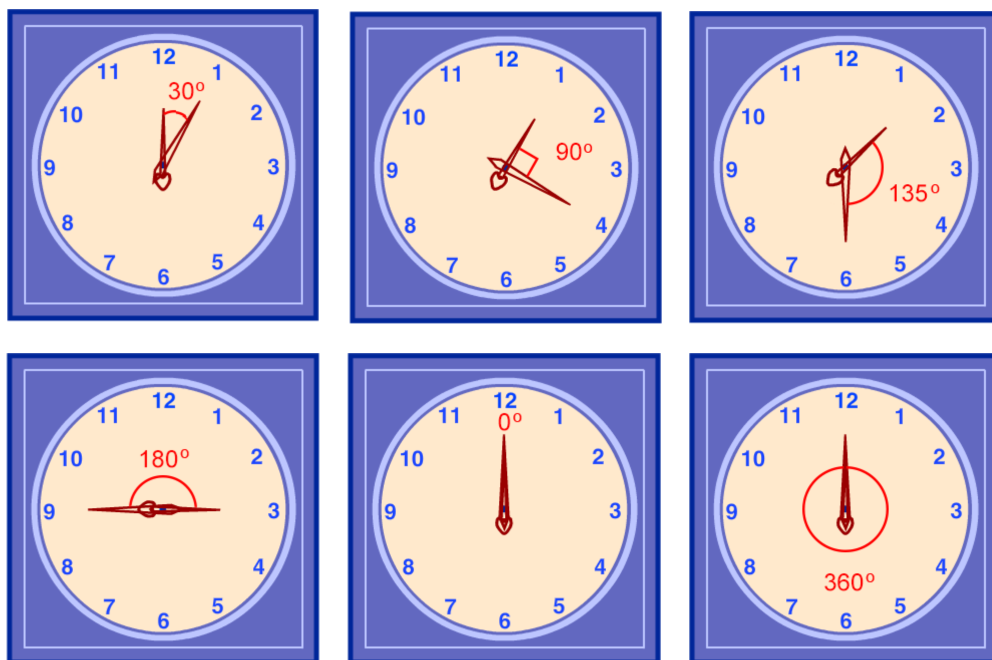
Hình 3.1.6. Thêm số cho đồng hồ

BÀI 2 GIỚI THIỆU VỀ GÓC VÀ HƯỚNG CỦA ĐỐI TƯỢNG

2.1. Giới thiệu về góc

Cứ hai kim bất kì trong đồng hồ đều sẽ tạo thành một góc. Đơn vị để đo độ lớn của góc là độ, kí hiệu ^{o}. Ví dụ 30^o (30 độ). Góc cũng dùng để mô tả chuyển động xoay. Góc xoay càng rộng thì giá trị góc càng lớn. Xoay hết một vòng tròn thì góc bằng 360^o.

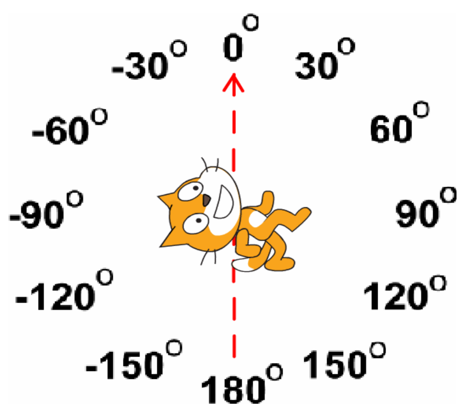
Hình 3.2.1 minh hoạ các giá trị góc được tạo bởi kim giờ và kim phút của đồng hồ. Lưu ý có thể có một số trường hợp không đúng với thực tế.



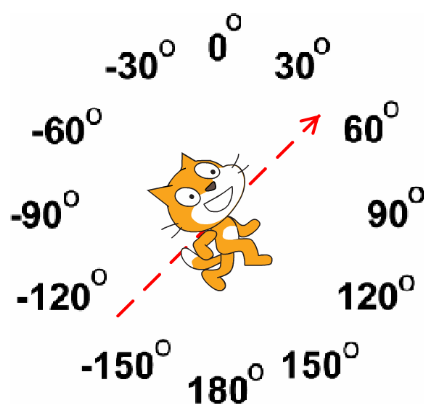
Hình 3.2.1. Các góc được tạo bởi kim giờ và kim phút

2.2. Hướng của đối tượng

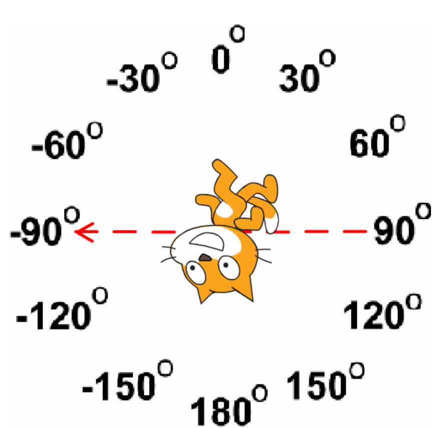
Mỗi đối tượng đều có một hướng nhất định và có giá trị ngầm định là hướng sang phải (90^o). Hướng và giá trị hướng của đối tượng được mô tả như trên *Hình 3.2.2*.



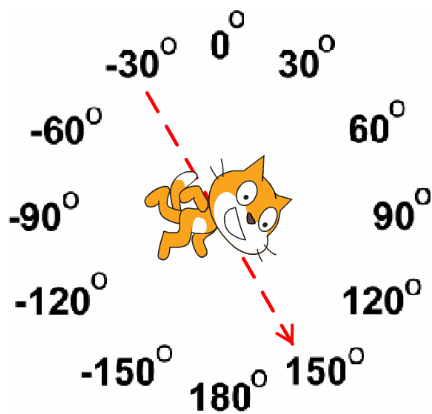
Hướng: 0°



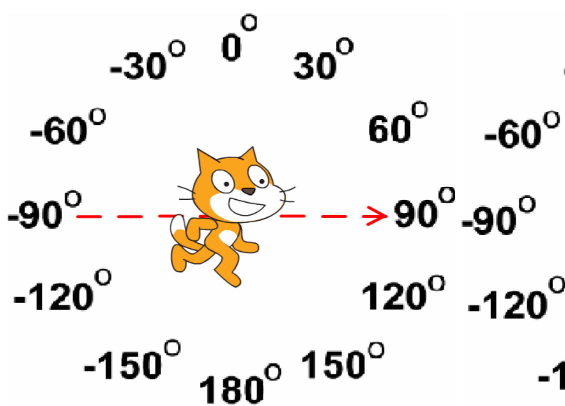
Hướng: 45°



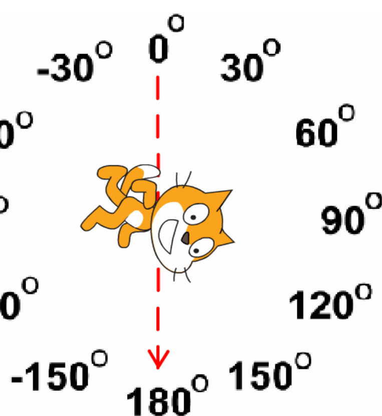
Hướng: -90°



Hướng: 150°



Hướng: 90°



Hướng: 180°

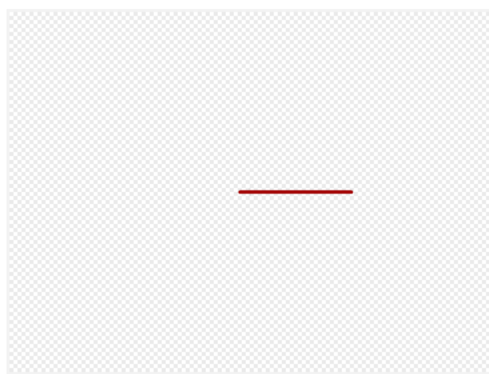
Hình 3.2.2. Một số ví dụ về hướng của đối tượng

BÀI 3 VẼ KIM ĐỒNG HỒ

Sau khi vẽ được mặt đồng hồ, các bạn cần thiết kế các kim đồng hồ và chỉnh lại vị trí các số cho chính xác.

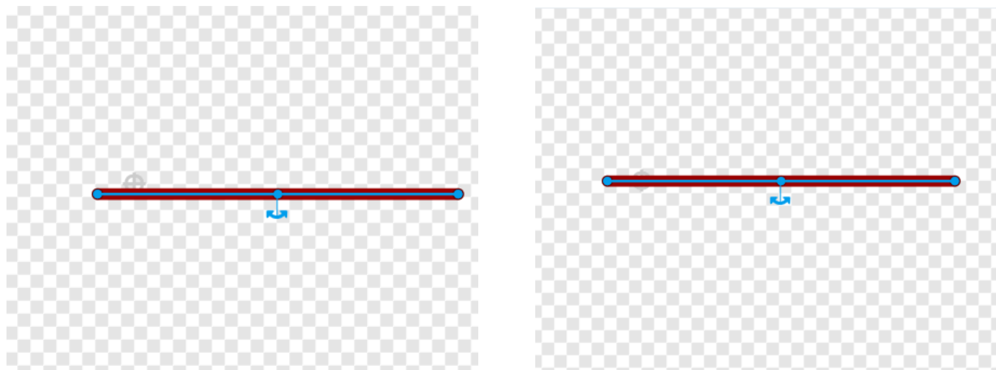
3.1. Kim giây

Hình dạng kim giây rất đơn giản, chỉ là một đường thẳng. Các bạn dùng công cụ ╱ ,chỉnh màu và kích thước nét vẽ rồi sau đó tiến hành vẽ. Khi vẽ nhấn giữ phím **Shift**.



Hình 3.3.1. Vẽ kim giây

Trên mặt đồng hồ, điểm quay của các kim không nằm ở chính giữa kim mà sẽ lệch về một bên, do đó các bạn cần chỉnh tâm của kim cũng lệch tương ứng như vậy.



Hình 3.3.2. Chỉnh tâm của kim

3.2. Kimphút

Cách vẽ kim phút phức tạp hơn một chút. Sau khi tạo đối tượng mới, các bạn vẽ đường viền hình chữ nhật, sau đó tạo hình bằng công cụ chỉnh hình.

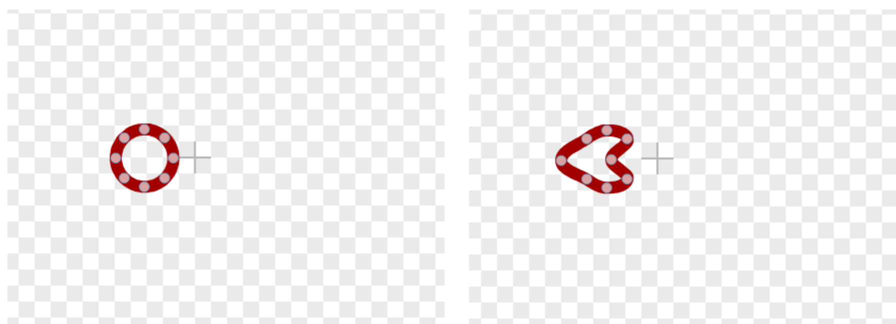
Khi ở chế độ chỉnh hình có các chấm tròn xuất hiện trên đối tượng, nháy chuột vào một điểm để bỏ điểm đó, nháy chuột vào một cạnh để tạo thêm một điểm.



Hình 3.3.3. Tạo hình kim phút

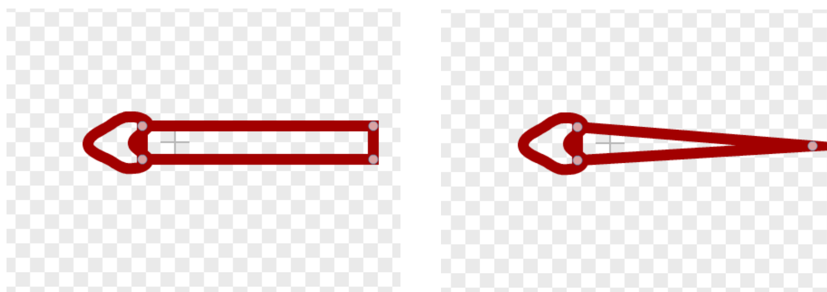
3.3. Kimgiờ

Đầu tiên vẽ một đường tròn, chỉnh hình để thành hình trái tim.



Hình 3.3.4. Tạo hình đầu kim giờ

Thêm một hình chữ nhật và chỉnh lại hình để tạo thành đầu kim.



Hình 3.3.5. Kim giờ Lưu ý: Sau khi vẽ xong, đổi lại tên cho các đối tượng.

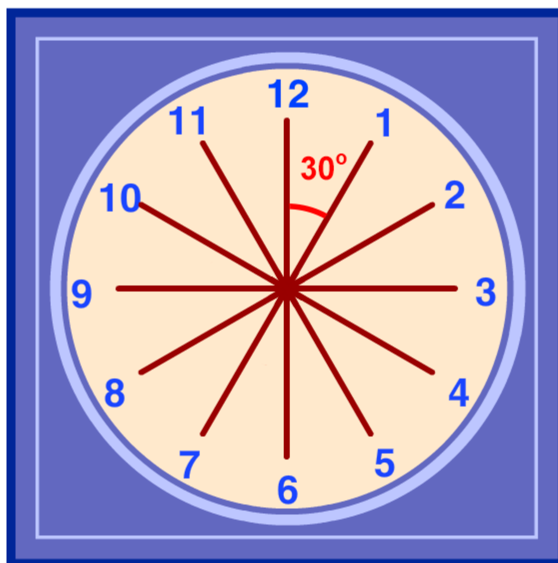


3.4. Chỉnh lại vị trí các số

Ở bài trước, các bạn đã thêm số cho mặt đồng hồ, tuy nhiên vị trí mới chỉ ước lượng, chưa chính

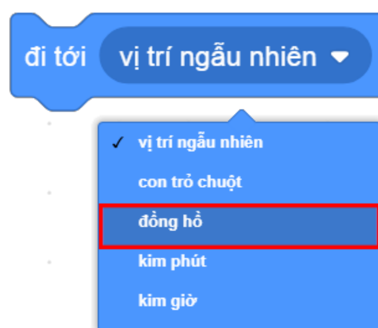
xác.

Nối 12 số với tâm sẽ được 12 góc, mỗi góc bằng 30°. Các bạn sẽ lập trình để kim giây xoay 12 lần, mỗi lần quay 30° và tạo bản sao. Sau đó sắp xếp lại vị trí các số trên mặt đồng hồ.



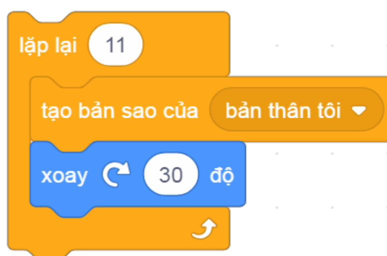
Hình 3.3.6. Vị trí các số

Đầu tiên lập trình để kim giây nhảy tới tâm của đồng hồ. Ở khu lập trình của kim giây, chúng ta sử dụng khối lệnh đi tới vị trí ngẫu nhiên và chọn đối tượng **đồng hồ**.



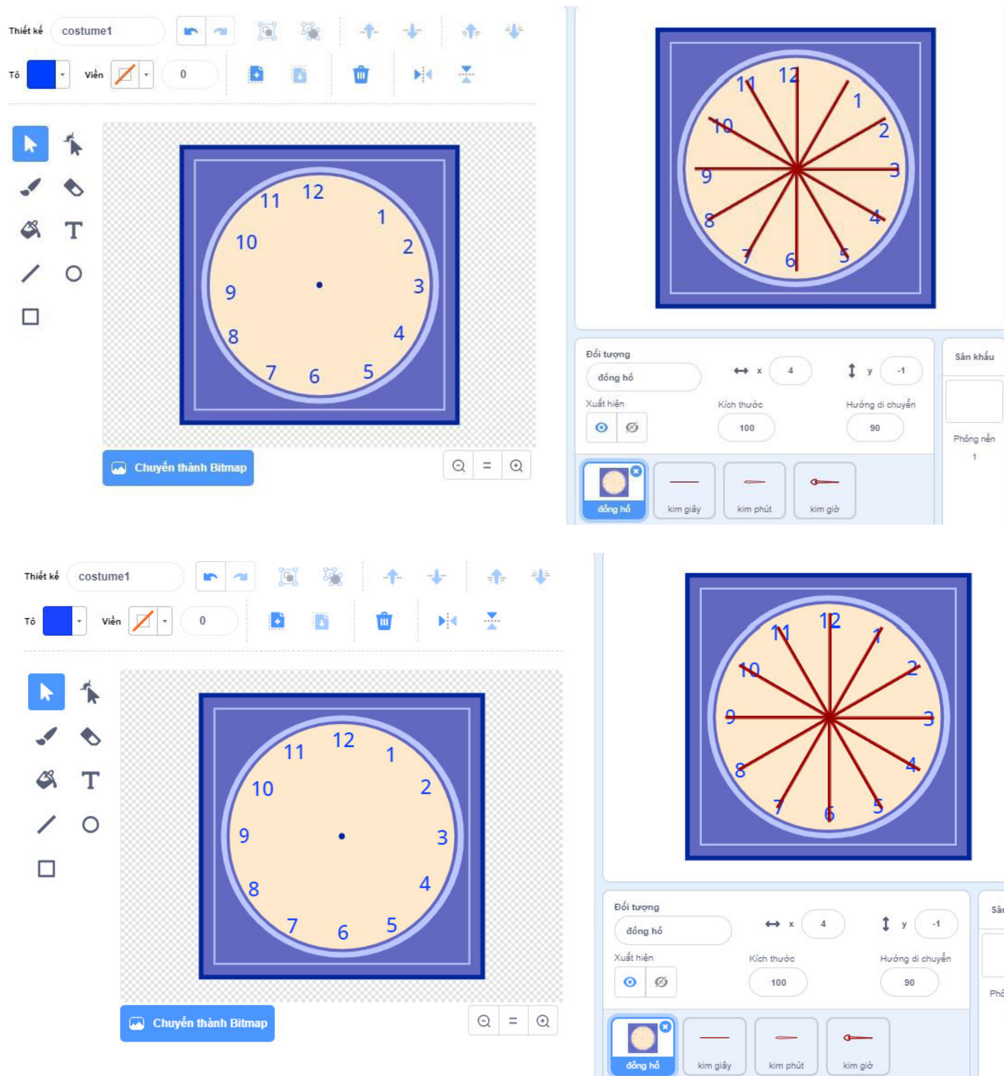
Hình 3.3.7. Kim giây đi tới tâm của đồng hồ

Tiếp theo, lập trình cho kim giây xoay và tạo bản sao.



Hình 3.3.8. Kim giây tạo bản sao

Cuối cùng, vào trang **Thiết kế** của đối tượng **đồng hồ** để sắp xếp lại vị trí các số.



Hình 3.3.9. Sắp xếp vị trí các số của đồng hồ

BÀI 4 LẬP TRÌNH KIM GIÂY

4.1. Hướng của đối tượng

Ngầm định hướng của đối tượng là hướng 90^{o}, tức là hướng sang phải. Các bạn có thể điều khiển để đối tượng hướng về một phía nhất định bằng khối lệnh hướng về phía 90 . Chúng ta có thể thay đổi giá trị góc xoay tuỳ ý. Giá trị góc này chính là hướng của nhân vật đã được nhắc đến ở Bài 2 và được minh hoạ ở *Hình 3.2.2.*

4.2. Lập trình kim giây

Khối lệnh ngày hiện tại chứa đầy đủ thông tin về ngày, tháng, năm, giờ, phút, giây của đồng hồ trên máy tính. Chạy thử khối lệnh và ta sẽ thấy được giá trị hiện tại của thời gian tương ứng.

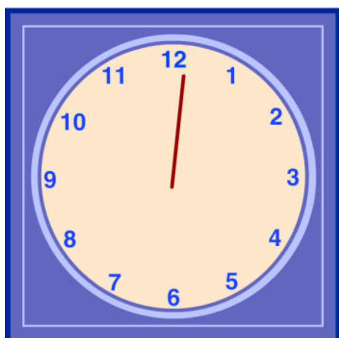


Hình 3.4.1. Khối lệnh thời gian với năm hiện tại và ngày hiện tại của máy tính

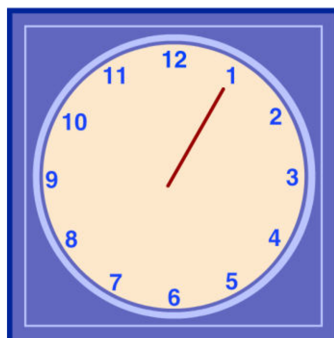
Sau khi đã biết được giây hiện tại bằng cách chọn **giây** trong khối lệnh giây hiện tại, chúng ta cần tính toán xem hướng mà kim giây phải xoay ứng với mỗi giây.

Để quay hết một vòng tròn (360^{o}) kim giây phải quay 60 lần, vậy mỗi lần kim giây sẽ quay một góc có giá trị bằng $360^{o}/60 = 6^{o}$.

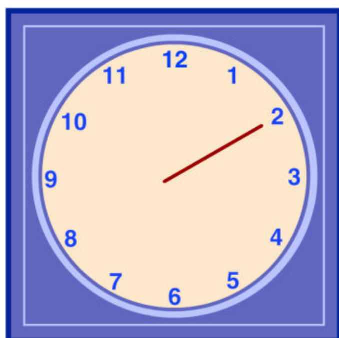
Giây đầu tiên, kim quay về hướng góc 6^{o}, giây tiếp theo là 12^{o} và cứ như vậy.



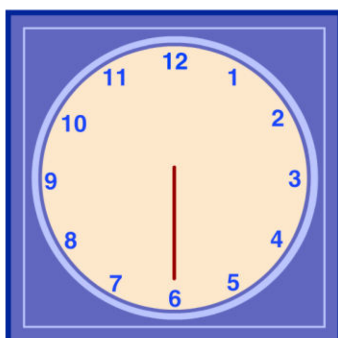
Giây thứ 1
Hướng: 6°



Giây thứ 5
Hướng: 30°



Giây thứ 10
Hướng: 60°



Giây thứ 30
Hướng: 180°

Hình 3.4.2. Hướng của kim giây

Vậy giá trị góc của hướng sẽ bằng giây hiện tại nhân với 6, ghép lại bằng cách kéo thả khối lệnh 

lên khối lệnh 

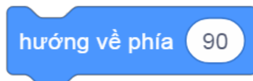
khi nổi viền trắng thì thả chuột, các bạn sẽ có được giá trị góc cần phải xoay 

.

Lưu ý: Di chuyển khối lệnh 

vào ô bên trái của khối lệnh 

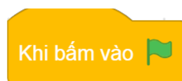
.

Ghép giá trị hướng vừa tính được vào khối lệnh xoay 

để điều khiển kim giây xoay về đúng góc.

Việc xoay này phải luôn được thực hiện, vì vậy khối lệnh 

sẽ được sử dụng.

Các bạn nhớ ghép vào dưới khối lệnh 

để kim có thể hoạt động khi chúng ta nháy lệnh .



Hình 3.4.3. Đoạn khối lệnh chuyển động của Kim giây

BÀI 5 LẬP TRÌNH KIM PHÚT

Để quay hết một vòng tròn (360°) kim phút phải quay 60 lần, vậy mỗi lần kim phút sẽ quay một góc có giá trị bằng 360°/60 = 6° và giống với kim giây. Vậy chúng ta chỉ cần lập trình giống với kim giây, nhưng thay vì lấy giá trị giây (giây hiện tại) các bạn sẽ lấy giá trị phút hiện tại (phút hiện tại).



Đoạn khối lệnh chuyển động của kim phút

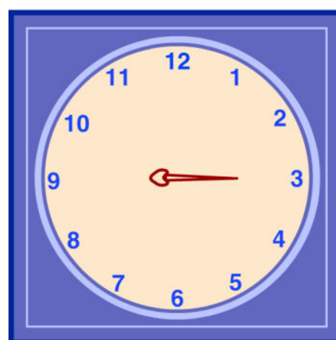
BÀI 6 LẬP TRÌNH KIM GIỜ

Kim giờ quay hết một vòng 360° ứng với 12 giờ, vậy mỗi giờ kim giờ phải xoay thêm một góc bằng 360°/12 = 30°.

Vậy thì giá trị góc của hướng ứng với giờ hiện tại sẽ bằng giờ hiện tại (giờ ▾ hiện tại) nhân với 30 là ((giờ ▾ hiện tại) * 30).



1 giờ
Hướng: 30°



3 giờ
Hướng: 90°



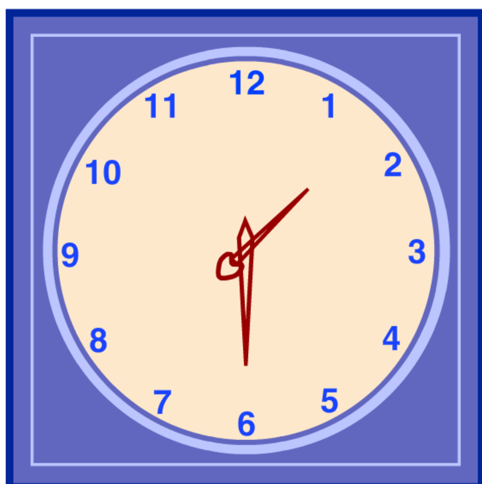
5 giờ
Hướng: 150°



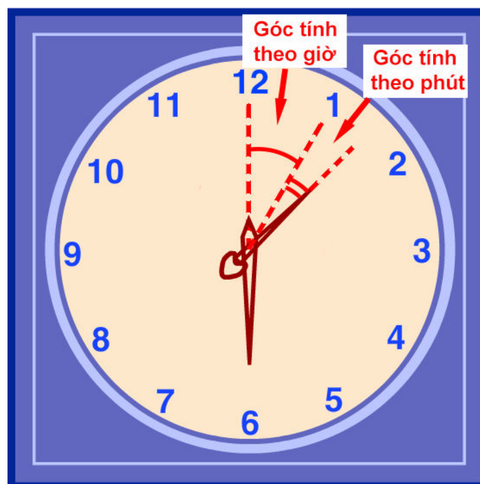
9 giờ
Hướng: 270° (-90°)

Hình 3.6.1. Hướng của kim giờ

Tuy nhiên, ở chiếc đồng hồ thật thì khi chuyển giữa các giờ, kim giờ sẽ xoay từng chút một chứ không xoay đột ngột mỗi lần 30°, tức là giá trị góc của hướng sẽ bằng góc tính theo giờ hiện tại cộng với góc tính theo phút hiện tại.



1 giờ 30 phút

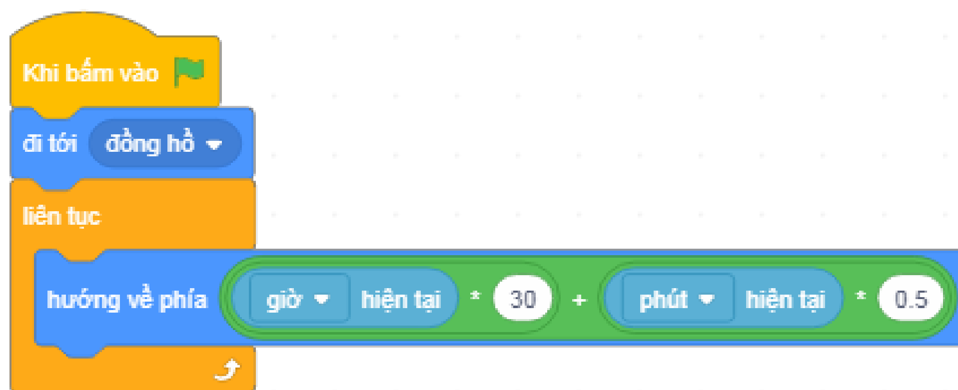


1 giờ 30 phút

Hình 3.6.2. Hướng của kim giờ tính thêm góc theo phút

Khi kim phút quay một vòng đủ 60 phút (hết một giờ) thì kim giờ sẽ xoay thêm một góc 30^0, vậy mỗi phút kim giờ sẽ xoay một góc 30^o chia 60 bằng 0.5^o.

Vậy góc tính theo phút sẽ bằng phút hiện tại [phút ▾ hiện tại] nhân với 0.5 là [(phút ▾ hiện tại) * 0.5] và tổng hai góc đó sẽ là hướng của kim giờ.



Hình 3.6.3. Đoạn khối lệnh chuyển động của kim giờ

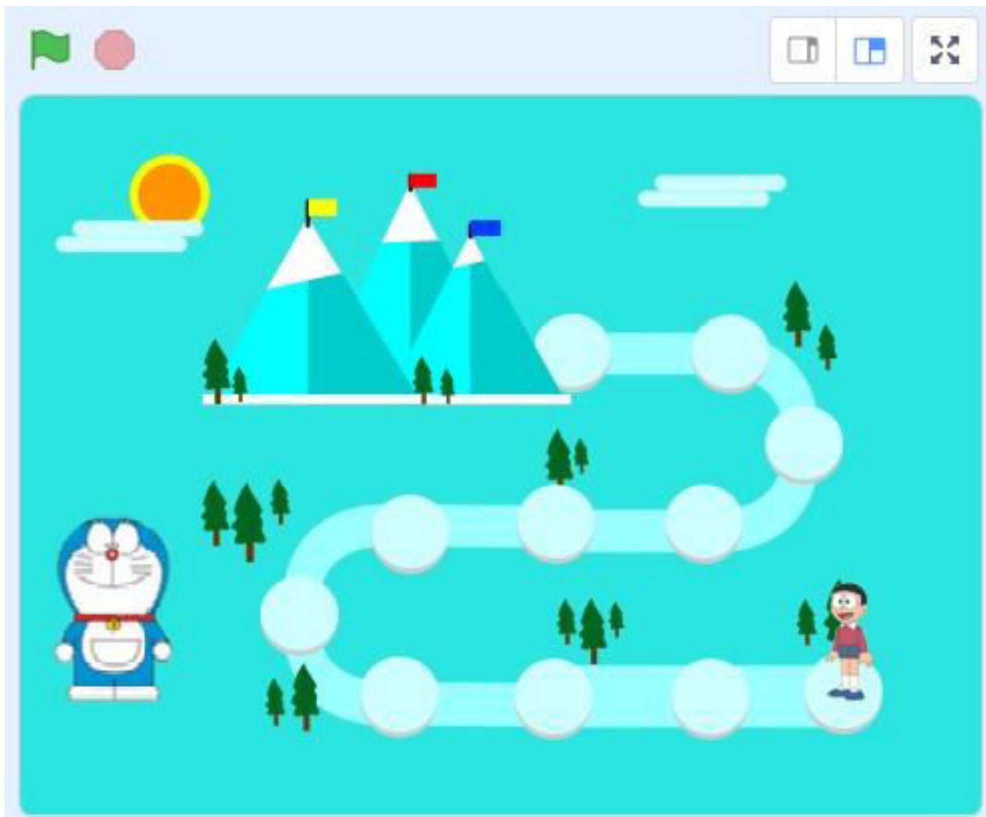
Chiếc đồng hồ đến đây đã được hoàn thành, các bạn có thể sáng tạo thêm bằng cách thêm quả lắc hoặc thêm chức năng hẹn giờ cho đồng hồ. Chúc các bạn thành công!

CHƯƠNG IV

Đường lên đỉnh Olympia

Trong chương này, các bạn sẽ được hướng dẫn để tạo ra một trò chơi rất hấp dẫn. Trong trò chơi này, các bạn sẽ đóng vai Nobita trong bộ phim hoạt hình Doraemon trả lời các câu hỏi là các phép tính để tiến lên đỉnh Olympia.

Tổng quan chương trình



Trong trò chơi, người chơi sẽ phải liên tục trả lời các câu hỏi toán học do nhân vật Doraemon đưa ra. Với mỗi câu trả lời đúng, nhân vật Nobita sẽ được tiến lên một bước. Để giành được chiến thắng, người chơi sẽ phải đưa Nobita lên đến đỉnh núi.

Chương trình gồm có các nhân vật, đối tượng:

1. Doraemon

Nhân vật Doraemon sẽ liên tục đưa ra các câu hỏi, người chơi nhập câu trả lời từ bàn phím.

2. Nobita

Mỗi khi người chơi trả lời đúng một câu hỏi của Doraemon, nhân vật Nobita sẽ được tiến lên một bước. Nếu Nobita tiến được đến đỉnh núi thì người chơi sẽ giành chiến thắng.

Một số chức năng của chương trình như thêm âm thanh, thay đổi hình dạng Nobita khi đến đích… sẽ không được hướng dẫn mà sẽ để trong phần Bài tập thực hành ở cuối mỗi bài. Các bạn cần làm bài tập đầy đủ để hoàn thiện chương trình.

BÀI 1 VẼ ĐƯỜNG LEO NÚI

Ở những bài đầu tiên, các bạn sẽ được hướng dẫn để tạo bản đồ trò chơi.

1.1. Vẽ đường leo núi

Trong phần tài liệu của bài học đã có sẵn hình nền là hình bản đồ đơn giản. Sau khi tải về và thêm vào chương trình, các bạn sẽ phải vẽ thêm đường leo núi vào bản đồ.

Hình 4.1.1. Bản đồ đơn giản



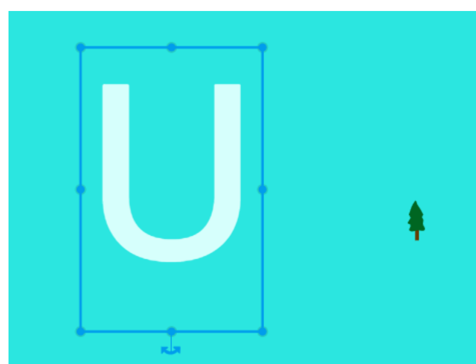
Trong khung nền sân khấu, nháy vào biểu tượng tải lên để đưa hình vào, chuyển đến khu thiết kế phông nền (Backdrop) bản đồ đã có sẵn một cái cây.

Đường leo núi gồm 2 chữ U quay vào nhau và lệch nhau một đoạn. Ta sử dụng công

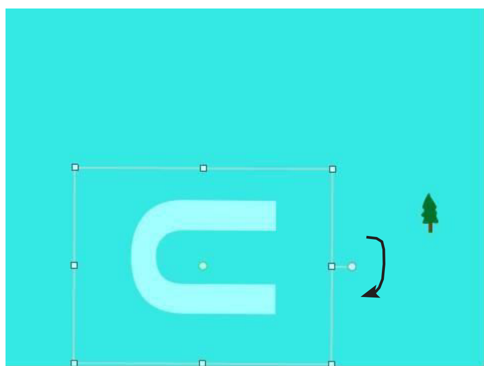
cụ **T** để viết một chữ U, sau đó phóng to và xoay ngang là được một nửa đường. Chữ U còn lại ta tạo bản sao từ chữ U đầu tiên rồi xoay ngược lại và sắp xếp lại. Thêm hình chữ nhật và hoàn thiện đường theo hình sau:



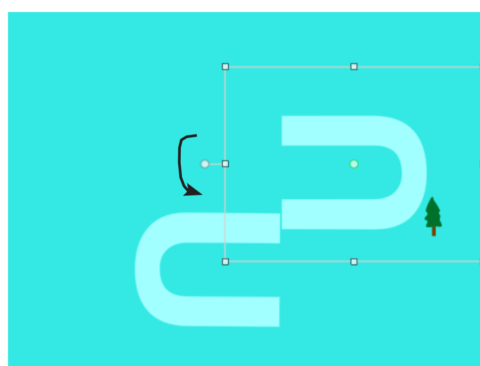
B1. Viết chữ



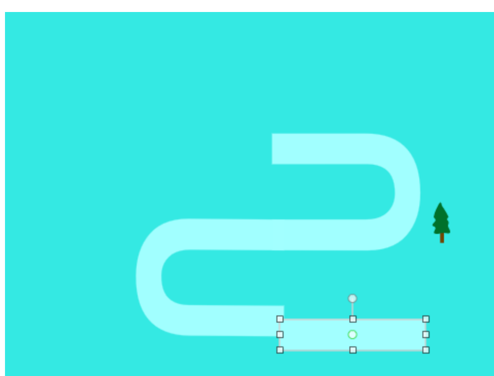
B2. Phóng to



B3. Xoay ngang



B4. Sao chép và xoay ngược lại

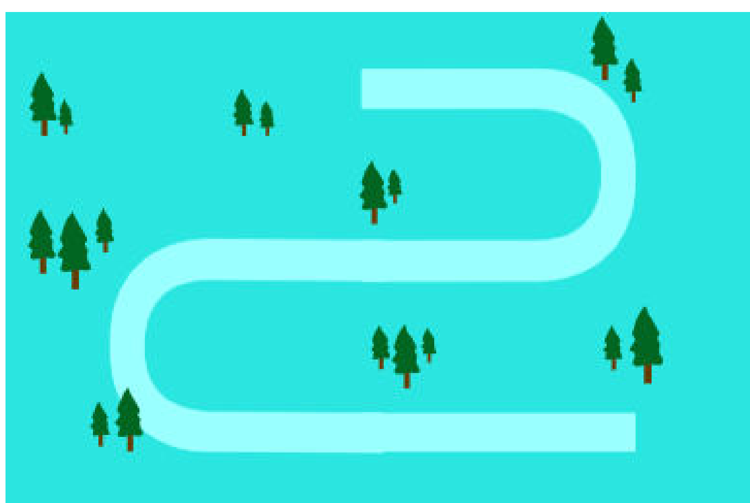


B5. Thêm hình chữ nhật



B6. Sắp xếp lại để có đường Hình 4.1.2. Các bước vẽ đường

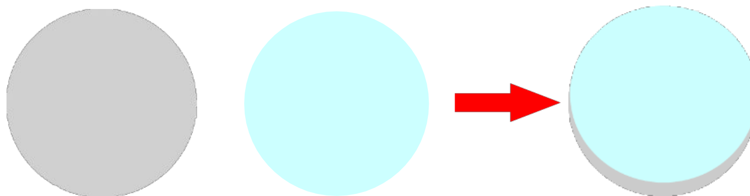
Sao và dán hình cái cây để trang trí cho đường đi.



Hình 4.1.3. Trang trí

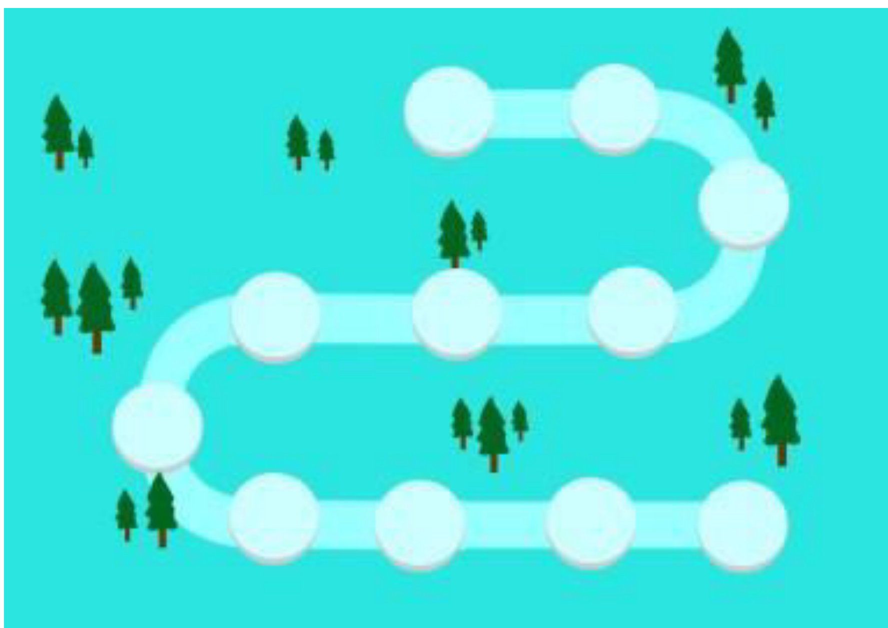
1.2. Vẽ bậc

Ta sẽ xây các bậc trên đường đi của nhân vật Nobita bằng cách sử dụng công cụ vẽ . Bậc gồm một hình tròn màu sáng và một hình tròn màu xám. Ta đặt hình tròn màu sáng lên trên hình tròn xám và sắp xếp hơi lệch nhau để hai hình tròn tạo thành một bậc.



Hình 4.1.4. Vẽ bậc

Sử dụng công cụ và để tạo bản sao một bậc thành nhiều bậc và sắp xếp như hình sau:

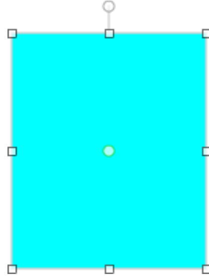


Hình 4.1.5. Xây bậc trên bản đồ

BÀI 2 VẼ NÚI VÀ MÂY

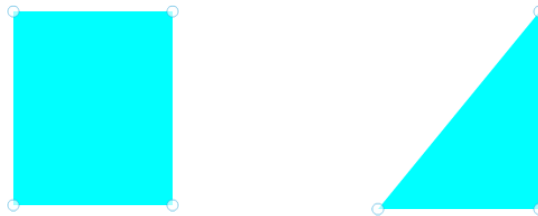
2.1.Vẽ núi

Đầu tiên vẽ một hình chữ nhật, ta có thể phóng to trang vẽ để có thể vẽ dễ dàng hơn.



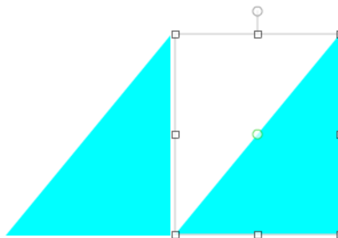
Hình 4.2.1. Vẽ hình chữ nhật

Sử dụng công cụ **Chỉnh hình** , nháy vào một điểm ở góc trên bên trái để nó trở thành hình tam giác. Đây chính là một nửa của ngọn núi.



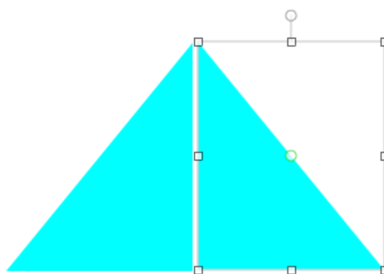
Hình 4.2.2. Vẽ nửa ngọn núi thứ nhất

Sử dụng công cụ và , tạo thêm một bản sao để vẽ nửa còn lại.



Hình 4.2.3. Tạo bản sao nửa ngọn núi

Sau đó nháy vào nút để lật ngược hình thứ hai vừa tạo.



Hình 4.2.4. Lật và sắp xếp lại nửa ngọn núi

Để ngọn núi có một bên chiếu sáng một bên bị che khuất, ta đổ màu tối hơn cho nửa bên phải.



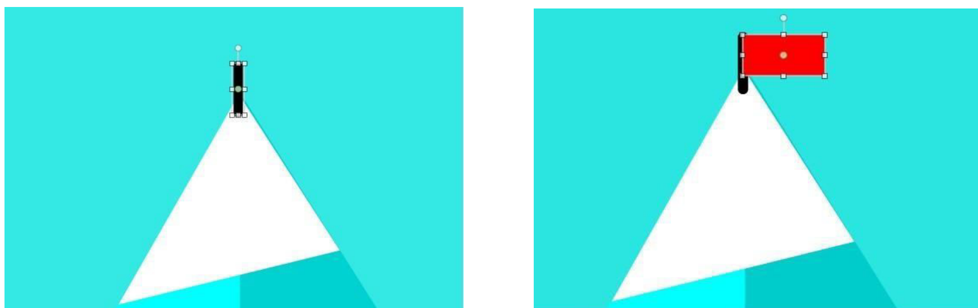
Hình 4.2.5. Đổ màu cho nửa bên phải

Tiếp đó, để trên đỉnh núi có tuyết phủ trắng, ta vẽ một hình chữ nhật màu trắng rồi dùng công cụ **Chỉnh hình** để chỉnh thành đỉnh núi.



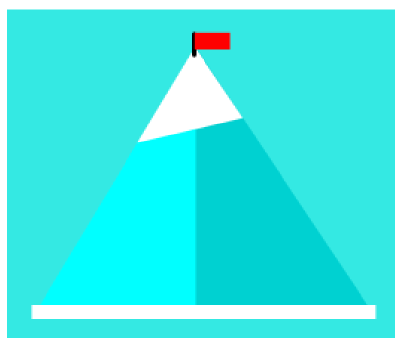
Hình 4.2.6. Vẽ tuyết trên đỉnh núi

Để vẽ lá cờ trên đỉnh núi, ta sử dụng công cụ **Vẽ đường thẳng** để vẽ cột cờ và công cụ **Hình chữ nhật** với màu nền màu đỏ để vẽ lá cờ.



Hình 4.2.7. Vẽ lá cờ

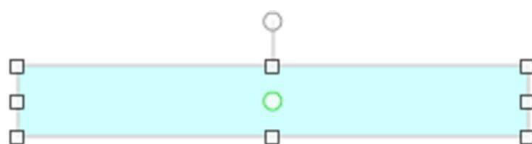
Cuối cùng, ta vẽ thêm một hình chữ nhật trắng dài dưới chân núi làm lớp tuyết. Nhấn giữ phím Ctrl, dùng công cụ chọn để chọn các chi tiết của núi, sau đó gộp nhóm . Chúng ta đã có một ngọn núi hoàn chỉnh.



Hình 4.2.8. Ngọn núi hoàn chỉnh

2.2. Vẽ mây

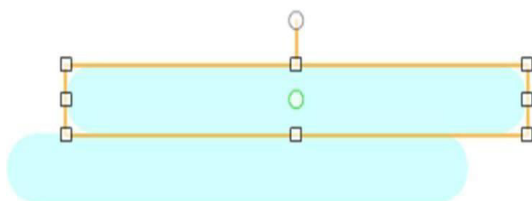
Mây của chúng ta khá đơn giản, gồm hai hình chữ nhật và bốn hình tròn. Các bước thực hiện được mô tả ở *Hình 4.2.9.*



B1. Vẽ hình chữ nhật



B2. Vẽ thêm hai hình tròn ở hai đầu



B3. Tạo bản sao



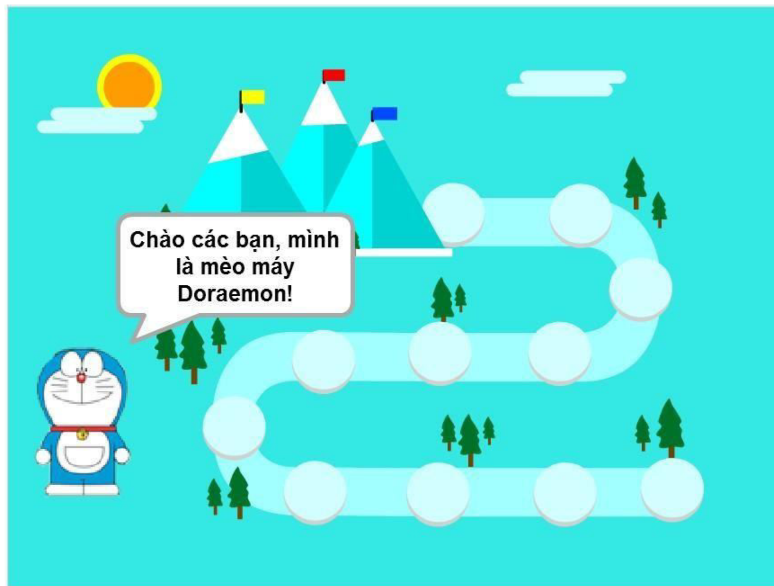
B4. Xếp lệch nhau Hình 4.2.9. Các bước vẽ mây

BÀI 3 LẬP TRÌNH CHO DORAEMON NÓI VÀ NGHĨ

3.1. Lập trình để nhân vật nói

Trước tiên, ta xoá nhân vật chú mèo có sẵn và tải tài liệu để thêm hình Doraemon vào **Khung nhân vật**.

Khi bắt đầu, chú mèo máy Doraemon sẽ giới thiệu về bản thân và chương trình.



Hình 4.3.1. Nhân vật nói

Để lập trình được cho Doraemon nói, ta sử dụng khối lệnh nói ( ) trong 2 giây hoặc nói ( ). Cả hai khối lệnh đều điều khiển đối tượng nói và có thể chỉnh sửa nội dung của lời nói tuỳ ý. Nhưng có sự khác biệt ở thời gian mà đối tượng nói.

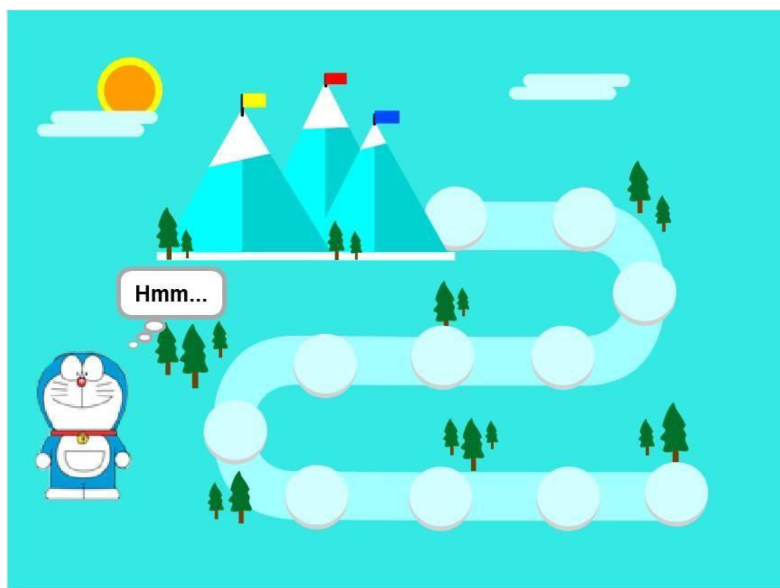
Với khối lệnh nói ( ) trong 2 giây, đối tượng sẽ nói trong một khoảng thời gian nhất định. Thời gian mặc định có sẵn trong khối lệnh là 2 giây, chúng ta có thể thay đổi lượng thời gian này tuỳ ý.

Với khối lệnh nói ( ), nhân vật sẽ hiện lời nói mãi cho đến khi ghép thêm khối lệnh nói lời khác hoặc ta nháy nút dừng.

3.2. Lập trình để nhân vật nghĩ

Để nhân vật nghĩ, chúng ta sử dụng hai khối lệnh nghĩ ( ) trong 2 giây và nghĩ ( ).

Sự khác nhau của hai khối lệnh này cũng tương tự khối lệnh nói.

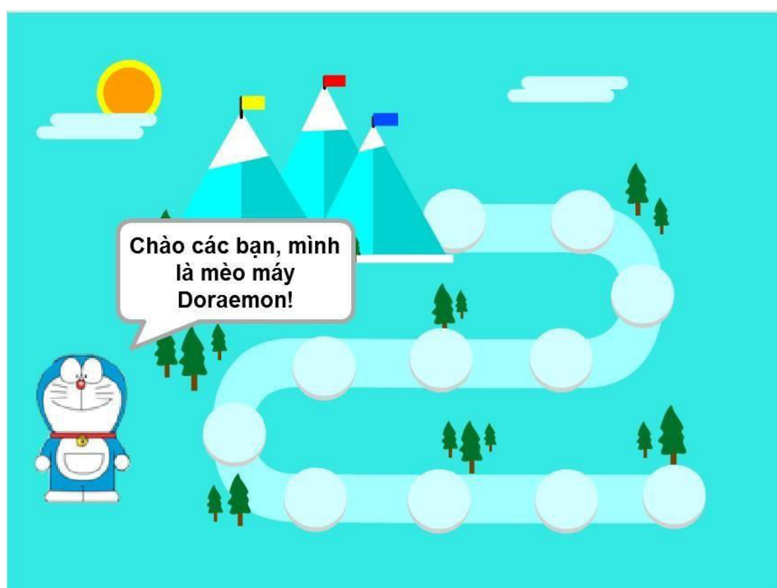


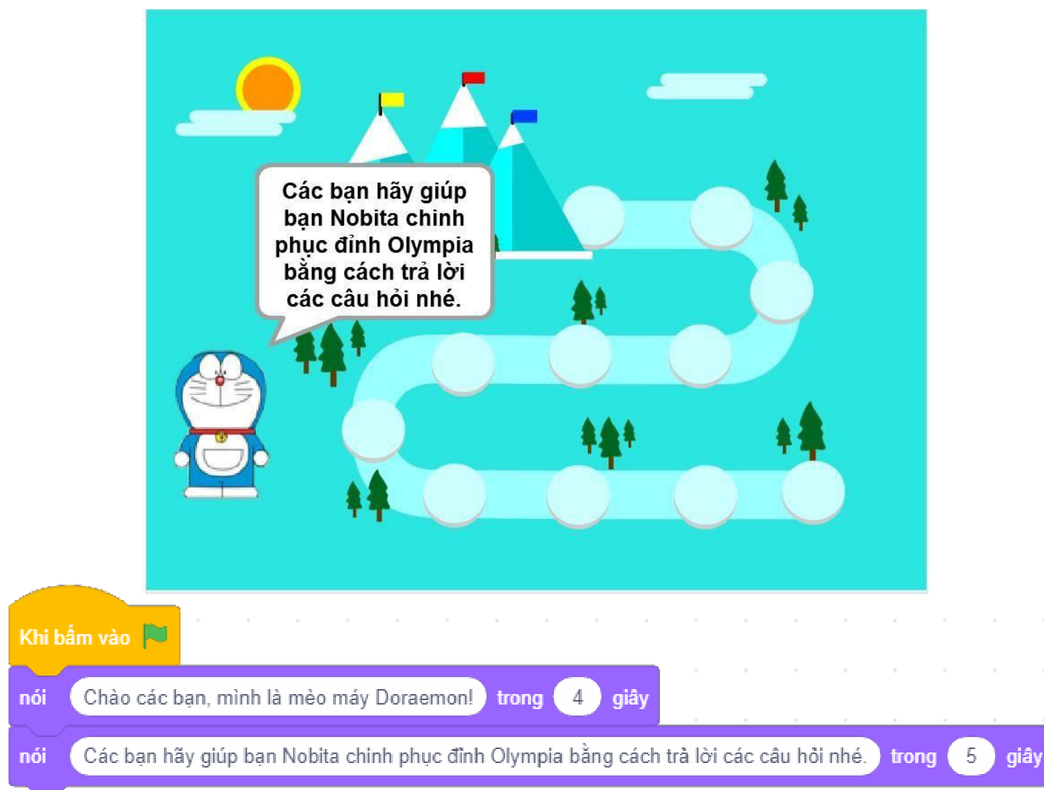
Hình 4.3.2. Nhân vật nghĩ

3.3. Lập trình cho Doraemon

Khi chạy chương trình, đầu tiên nhân vật Doraemon sẽ giới thiệu bản thân và chương trình trong một khoảng thời gian ngắn. Chúng ta sẽ sử dụng khối lệnh 

. *Hình 4.3.3* là một ví dụ mẫu cho lời giới thiệu.





Hình 4.3.3. Ví dụ Doraemon giới thiệu

BÀI 4 GIỚI THIỆU VỀ BIẾN

Biến là một khái niệm rất quan trọng. Trong bài này, các bạn sẽ được tìm hiểu về cách tạo biến và một số khối lệnh thao tác với biến trong Scratch.

4.1. Biến trong Scratch

Biến là một đại lượng được đặt tên dùng để lưu trữ số hoặc kí tự và có thể thay đổi được. Trong Scratch, biến là một khối lệnh có hình dạng , chứa một giá trị nhất định có thể là một số, một kí tự hoặc một chuỗi kí tự.



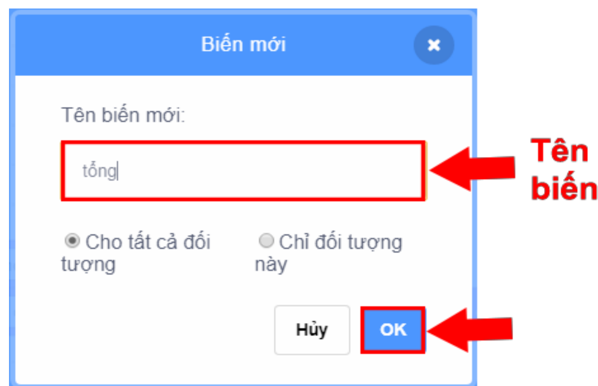
Hình 4.4.1. Các biến khác nhau với các giá trị của chúng

4.2. Cách tạo biến

Tất cả các thao tác với biến ở nhóm **Các biến số**. Tạo một biến mới bằng cách nháy vào nút



trong nhóm **Các biến số** và một cửa sổ sẽ hiện ra.

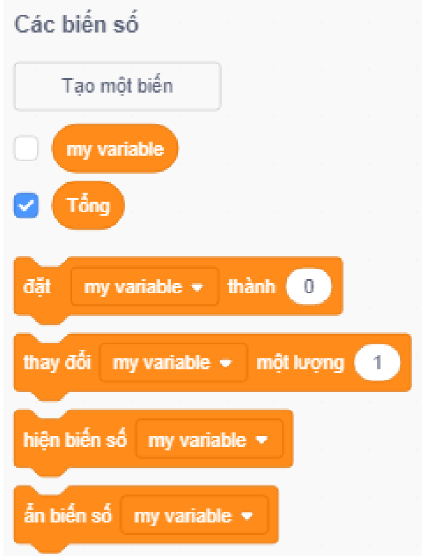


Hình 4.4.2. Cửa sổ tạo biến mới

Ta nhập tên biến vào ô *Tên biến mới*. Tên biến nên có ý nghĩa khơi gợi chức năng của biến. Ví dụ như nếu biến dùng để chứa giá trị tổng của hai số thì nên đặt tên là **Tổng**, biến để lưu điểm thì nên đặt tên là **Điểm**.

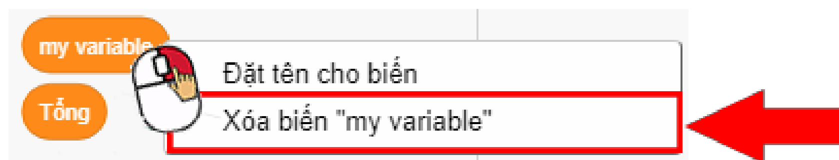
Trong cửa sổ tạo biến có hai lựa chọn. Lựa chọn *Chỉ đối tượng này* để biến chỉ được thao tác, sử dụng bởi đối tượng hiện tại đang lập trình. Lựa chọn *Cho tất cả đối tượng* để biến được tạo ra có thể được thao tác bởi tất cả các nhân vật khác trong chương trình. Nháy nút OK để tạo biến.

Trong ví dụ này, một biến tên là **Tổng** được tạo ra để tất cả các nhân vật đều có thể sử dụng được.



*Hình 4.4.3. Ví dụ biến **Tổng***

Mặc định trong danh sách biến ban đầu luôn có biến **my variable**. Các bạn có thể xóa biến **my variable** bằng cách nháy chuột phải vào khối lệnh biến trong danh sách biến, sau đó chọn lệnh *xóa biến "my variable"*



*Hình 4.4.4. **Xóa** biến*

4.3. Cách sử dụng biến

Sau khi tạo biến, những khối lệnh giúp thao tác với biến sẽ xuất hiện trên bảng danh mục các khối lệnh:



: Đặt cho biến **Tổng** giá trị bất kì. Trong trường hợp này giá trị được đặt cho biến **Tổng** là 0.



: Thay đổi biến **Tổng** một lượng bất kì. Trong trường hợp này lượng giá trị thay đổi của biến **Tổng** là 1.

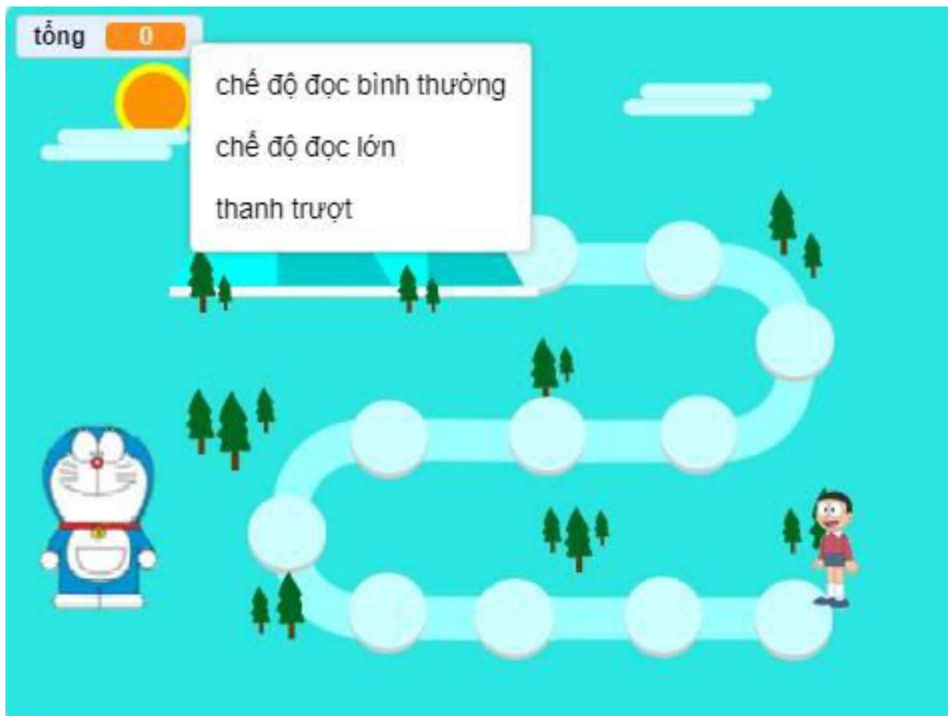


: Hiển thị biến **Tổng** lên trên sân khấu.



: Ẩn biến **Tổng** đi.

Khi hiển thị trên sân khấu, biến có ba kiểu để hiển thị. Để chuyển đổi kiểu hiển thị, ta nháy đúp chuột lên khung hiển thị của biến trên sân khấu, hoặc nháy nút phải chuột vào khung hiển thị và chọn cách hiển thị cho biến.



Hình 4.4.4. Chuyển đổi kiểu hiển thị biến

Các kiểu hiển thị:

0 : Chỉ hiển thị giá trị của biến.

tổng 0 Hiển thị tên của biến cùng với giá trị.

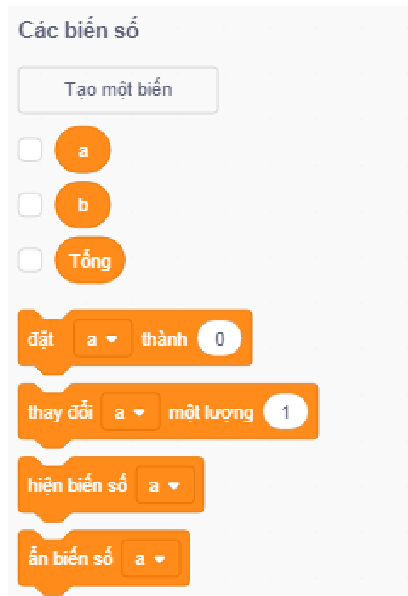
tổng 0 Hiển thị tên biến, giá trị của biến và thanh trượt. Ta có thể dùng thanh trượt để thay đổi giá trị của biến.

BÀI 5 LẬP TRÌNH CHO DORAEMON TÍNH TOÁN

Tính tổng hai số ngẫu nhiên

Một phép tính cộng gồm hai số hạng và tổng. Chúng ta đã có biến để chứa giá trị của tổng từ bài trước, vì thế nên ta chỉ cần tạo thêm hai biến *a* và *b* để lưu trữ giá trị của hai số hạng.

Hình 4.5.1. Thêm hai biến lưu hai số hạng



Doraemon sẽ đặt ra câu hỏi là tính tổng hai số ngẫu nhiên. Hai số hạng đều được lấy ngẫu nhiên, tức là chúng ta phải đặt cho biến *a* và biến *b* giá trị ngẫu nhiên. Để lấy giá trị ngẫu nhiên, chúng ta sẽ cần đến khối lệnh lấy ngẫu nhiên từ ( ) đến ( ) đã được hướng dẫn ở *Bài 6* trong *Chương 2*.

Ta kéo khối lệnh đặt biến và nháy vào mũi tên để chọn biến *a* và sau đó kéo khối lệnh lấy ngẫu nhiên từ ( ) đến ( ) vào đặt a thành ( ) để đặt cho biến *a* giá trị ngẫu nhiên và thay đổi giá trị ngẫu nhiên từ 1 đến 20.



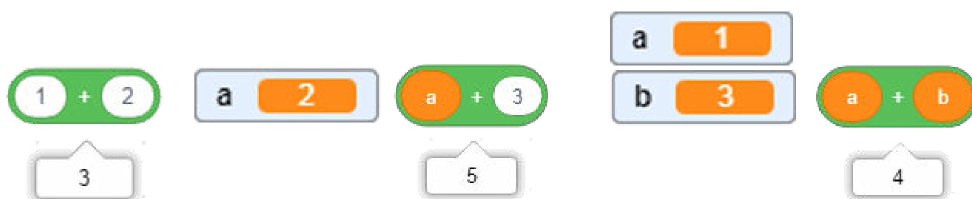
Hình 4.5.2. Đặt cho biến a giá trị ngẫu nhiên

Tạo bản sao và thay đổi biến *a* thành biến *b*, sau đó ghép hai khối lệnh lại với nhau như Hình 4.5.3.



Hình 4.5.3. Đặt hai biến giá trị ngẫu nhiên

Sau khi đã có hai số hạng, chúng ta sẽ lập trình để tính tổng hai số hạng này và lưu vào biến **Tổng**. Chúng ta sử dụng khối lệnh để tính tổng hai số. Hình 4.5.4 mô tả cách sử dụng khối lệnh này.



Hình 4.5.4. Khối lệnh tính tổng

Việc cuối cùng là đặt giá trị biến **Tổng** là giá trị của khối lệnh và ghép vào đoạn khối lệnh bên trên.



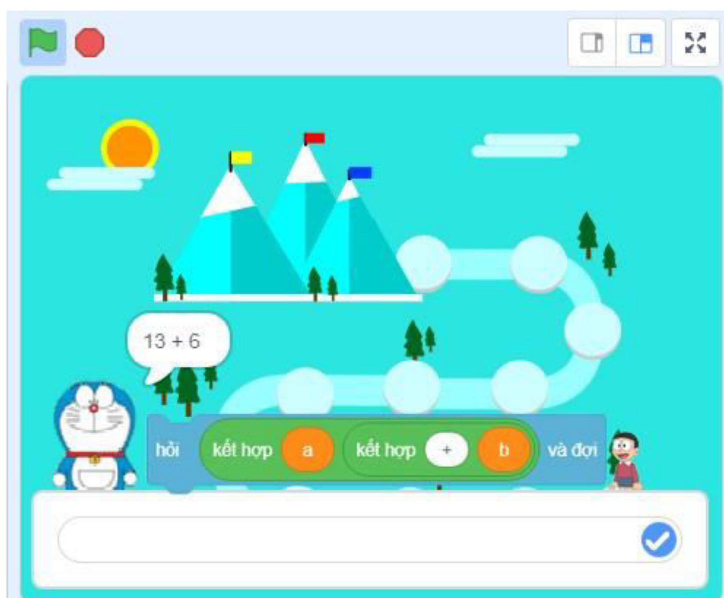
Hình 4.5.5. Đoạn khối lệnh tính toán

BÀI 6 LẬP TRÌNH ĐỂ DORAEMON ĐẶT CÂU HỎI

Sau khi đã tính toán xong, Doraemon sẽ đặt ra câu hỏi về phép tính tổng vừa được tính.

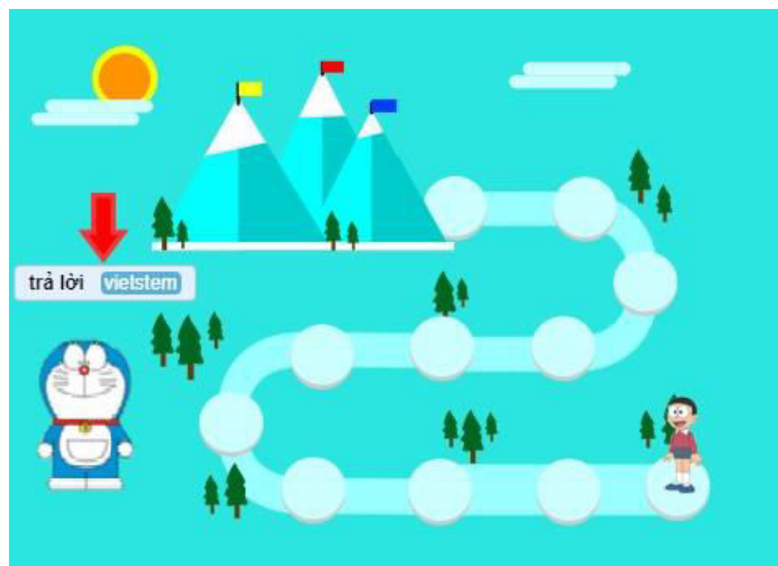
6.1. Khối lệnh hỏi

Để Doraemon đặt câu hỏi, ta sử dụng khối lệnh hỏi Tên của bạn là gì? và đợi trong nhóm **Cảm biến**. Ngầm định nội dung câu hỏi là "Tên bạn là gì ?". Nếu chạy thử thì ta sẽ thấy Doraemon hỏi và trên sân khấu xuất hiện thêm một khung để nhập câu trả lời.



Hình 4.6.1. Khối lệnh hỏi

Chỉ khi nào chúng ta nhập câu trả lời và nhấn phím Enter hoặc nháy nút ✓ thì Doraemon mới ngừng hỏi và câu trả lời vừa nhập vào sẽ được lưu lại ở khối lệnh trả lời trong nhóm **Cảm biến**. Ta có thể chạy khối lệnh để xem lại câu trả lời được lưu, hoặc nháy vào ô tích bên cạnh ☐ trả lời để chuyển thành ☑ trả lời và giá trị khối lệnh lưu sẽ hiển thị trên sân khấu.



Hình 4.6.2. Giá trị câu trả lời được lưu

6.2. Khối lệnh nối

Nội dung câu hỏi mà Doraemon hỏi là phép toán tính tổng hai biến *a* và *b*. Nếu ghép khối lệnh (a + b) vào thành nội dung câu hỏi thì Doraemon nói ra kết quả của tổng chứ không phải đưa ra phép toán, vậy nên chúng ta sẽ phải sử dụng khối lệnh (kết hợp táo chuối). Ở khối lệnh nối này đã có giá trị ngầm định là nối "hello " và world", các bạn chạy thử sẽ thấy cách khối lệnh hoạt động.

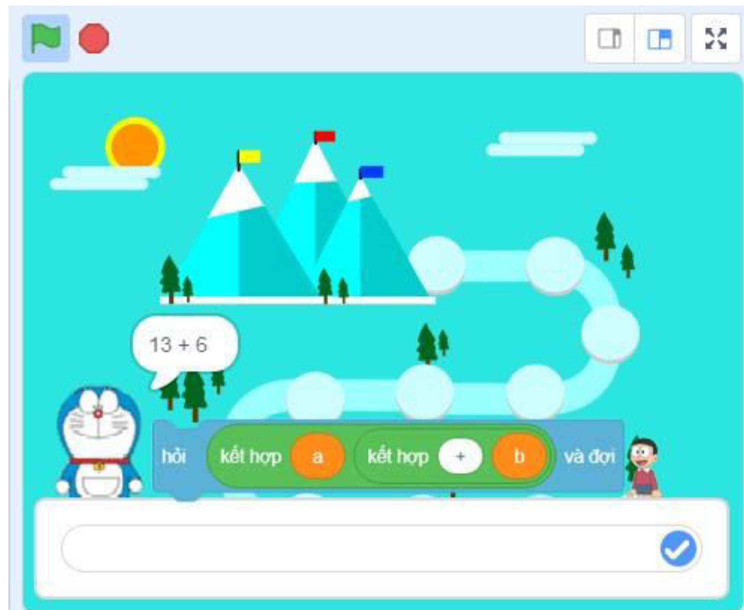


Hình 4.6.3. Khối lệnh nối

Trong trường hợp này, các bạn cần nối giá trị biến *a*, dấu „+" và giá trị biến *b* lại, tổng cộng là ba

nối. Chúng ta ghép nối các khối lệnh thành (kết hợp ( ) (kết hợp ( ) ( ))), sau đó ghép biến *a* và biến *b* vào hai bên và điền dấu „+" vào giữa, ta được

(kết hợp (a) (kết hợp (+) (b))).



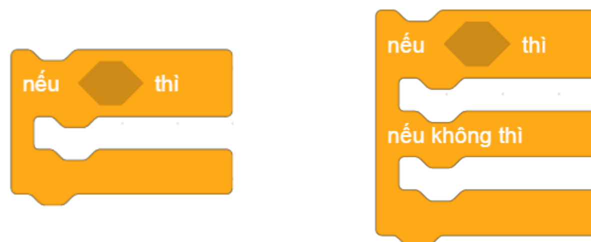
Hình 4.6.4. Kết quả

BÀI 7 LẬP TRÌNH ĐỂ DORAEMON KIỂM TRA CÂU TRẢ LỜI

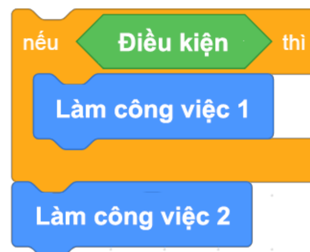
Khi Doraemon đặt câu hỏi, người chơi sẽ phải nhập câu trả lời vào và Doraemon sẽ kiểm tra câu trả lời, nếu đúng thì sẽ cho Nobita đến gần đích hơn, nếu sai thì sẽ nói "Sai rồi !". Bài học này sẽ hướng dẫn các bạn lập trình cho Doraemon thực hiện điều đó.

7.1. Hai khối lệnh kiểm tra điều kiện

Nhóm điều khiển có hai khối lệnh để kiểm tra điều kiện:



Cả hai khối lệnh đều có ô chứa khối lệnh giá trị hình lục giác (còn gọi là khối lệnh điều kiện, chứa khối lệnh dạng Điều kiện).



Hình 4.7.1. Khối lệnh Nếu – thì

Với khối lệnh **Nếu – thì**, nếu khối lệnh Điều kiện chứa giá trị **true** (đúng) thì đoạn khối lệnh Làm công việc 1 sẽ được thực hiện, nếu không thì bỏ qua và tiếp tục thực hiện khối lệnh tiếp theo Làm công việc 2.

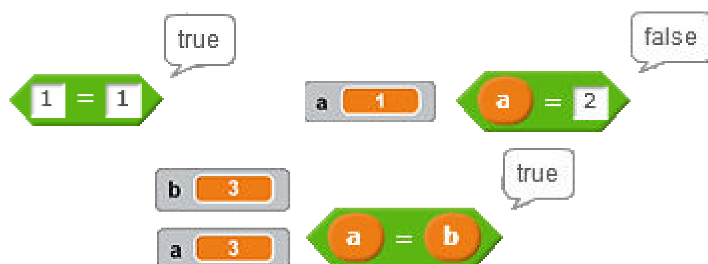


Hình 4.7.2. Khối lệnh Nếu – thì – nếu không thì

Với khối lệnh **Nếu – thì – nếu không thì**, nếu khối lệnh điều kiện Điều kiện chứa giá trị đúng thì khối lệnh Làm công việc 1 sẽ được thực hiện và bỏ qua khối lệnh Làm công việc 2. Nếu khối lệnh Điều kiện chứa giá trị sai thì sẽ ngược lại, bỏ qua khối lệnh Làm công việc 1 và thực hiện khối lệnh Làm công việc 2. Sau đó sẽ đều tiếp tục thực hiện các khối lệnh tiếp theo.

7.2. Kiểm tra câu trả lời

Doraemon sẽ kiểm tra xem câu trả lời có đúng với đáp án hay không, tức là sẽ phải so sánh đáp án với câu trả lời nhập vào có bằng nhau không. Để so sánh hai giá trị có bằng nhau hay không, chúng ta sử dụng khối lệnh (=).



Hình 4.7.3. Khối lệnh so sánh

Biến **Tổng** lưu đáp án còn khối lệnh trả lời lưu câu trả lời. Để so sánh hai giá trị này với nhau, ta ghép thành .

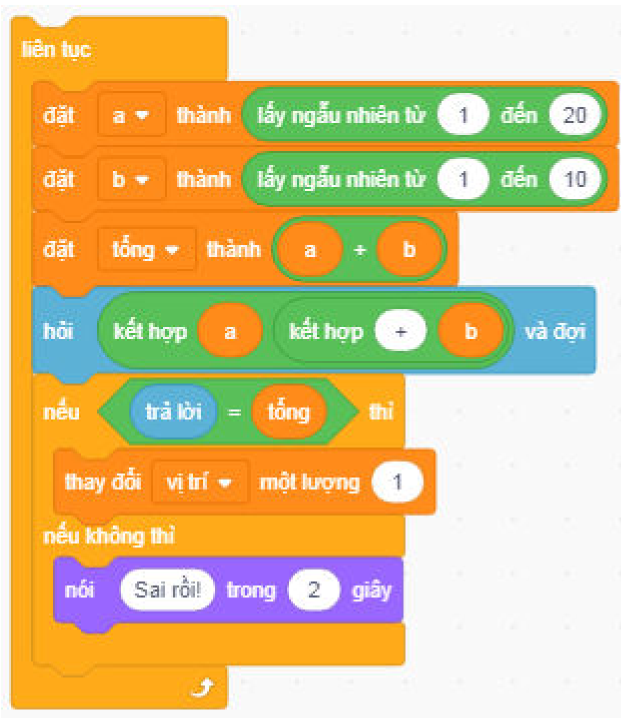
Nếu trả lời trả lời bằng với tổng, thì sẽ đưa Nobita đến gần đích hơn bằng cách thay đổi giá trị biến **vị trí**. Chúng ta sử dụng biến vị trí đã được thêm vào ở phần bài tập thực hành của bài trước để lưu vị trí của Nobita. Khi trả lời đúng thì vị trí của Nobita sẽ được tăng lên một bậc, ta lập trình cho biến **vị trí** tăng thêm 1 bằng khối lệnh thay đổi vị trí một lượng 1 Còn nếu sai thì Doraemon sẽ nói "Sai rồi !!!".



Hình 4.7.4. Kiểm tra câu trả lời

Sau khi trả lời xong, Doraemon sẽ hỏi lại câu hỏi khác, việc Doraemon đặt câu hỏi và kiểm

tra câu trả lời sẽ được lặp lại mãi.



Hình 4.7.5. Đoạn khối lệnh hoàn chỉnh

BÀI 8 LẬP TRÌNH CHO NOBITA

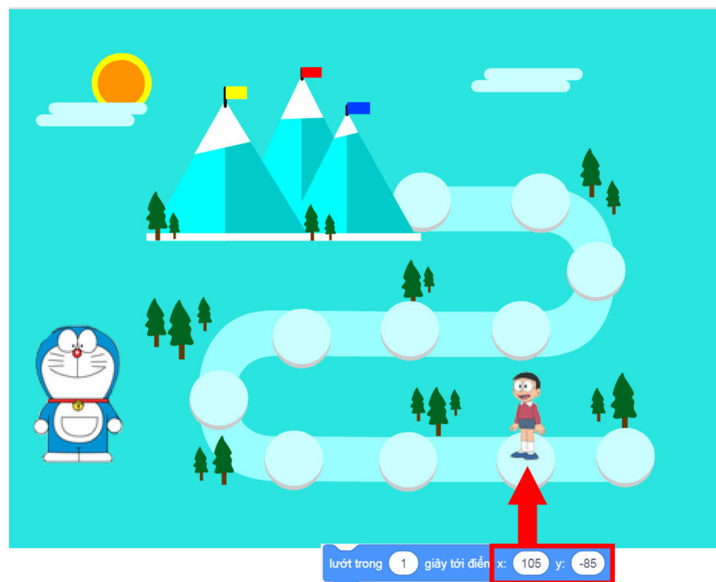
8.1. Khối lệnh Glide

Khối lệnh [lướt trong ( ) giây tới điểm x: ( ) y: ( )] là khối lệnh thuộc nhóm **Chuyển động**. Khối lệnh này giúp điều khiển nhân vật trượt tới vị trí nhất định trong một khoảng thời gian nhất định.



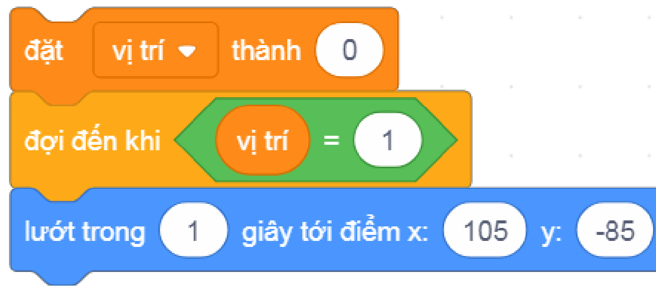
Hình 4.8.1. Khối lệnh Glide

Khi bắt đầu chương trình, Nobita sẽ đứng ở bậc đầu tiên và việc di chuyển đến bậc tiếp theo chúng ta sẽ sử dụng khối lệnh **Trượt**.



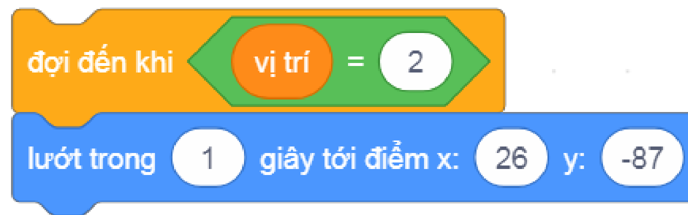
Hình 4.8.2. Nobita di chuyển

Ta chọn nhân vật Nobita, vào **Khu lập trình**, khởi tạo biến [vị trí] bằng 0, Nobita sẽ phải đợi cho đến khi người chơi nhập vào câu trả lời đúng, biến **vị trí** tăng thêm 1 thì mới di chuyển đến bậc thứ hai. Chúng ta sử dụng khối lệnh [đợi đến khi] trong nhóm **Điều khiển** để làm việc này.



Hình 4.8.3. Đoạn khối lệnh ứng với bậc đầu tiên

Để từ bậc thứ hai, Nobita di chuyển lên bậc thứ ba, ta cần xác định được toạ độ bậc thứ ba, đồng thời phải đợi cho đến khi người chơi trả lời đúng câu hỏi tiếp theo (biến **vị trí** tăng thêm 1 thành 2) mới tiến hành di chuyển. Tương tự với các bậc tiếp theo.



Hình 4.8.4. Đoạn khối lệnh ứng với bậc thứ hai

8.2. Khối lệnh xoay và điều chỉnh quay

Đến bậc thứ tư, Nobita phải xoay người hướng ngược lại (hướng sang phải), do đó các bạn điều khiển Nobita xoay một góc 180° bằng khối lệnh xoay ↻ 180 độ

Khi chạy thử, Nobita bị quay ngược xuống dưới. Để nhân vật không bị lật, ta sử dụng khối lệnh đặt kiểu xoay trái - phải tạo thành đoạn lệnh sau:



Hình 4.8.5. Thay đổi kiểu xoay của nhân vật

BÀI 9 KẾT THÚC CHƯƠNG TRÌNH

Bài học cuối cùng trong chương sẽ hướng dẫn các bạn hoàn thiện chương trình và cách dừng chương trình để kết thúc khi Nobita đã đến đích.

Hoàn thiện chương trình

Mỗi khi trả lời đúng, Nobita chưa kịp di chuyển lên vị trí mới Doraemon đã hỏi câu hỏi tiếp theo. Chúng ta lập trình để Doraemon đợi thêm một khoảng thời gian cho Nobita di chuyển rồi mới hỏi bằng khối lệnh đợi 2 giây .Ta thay đổi lệnh với nhân vật Doraemon như sau:



Hình 4.9.1. Hiệu chỉnh Doraemon

Khi Nobita đến đích, chương trình chưa được dừng lại và Doraemon vẫn còn hỏi tiếp. Để kết thúc chương trình, ta sử dụng khối lệnh dừng lại tất cả trong nhóm **Điều khiển**. Ta thay đổi lệnh với nhân vật Doraemon như sau:



Hình 4.9.2. Ngừng toàn bộ chương trình

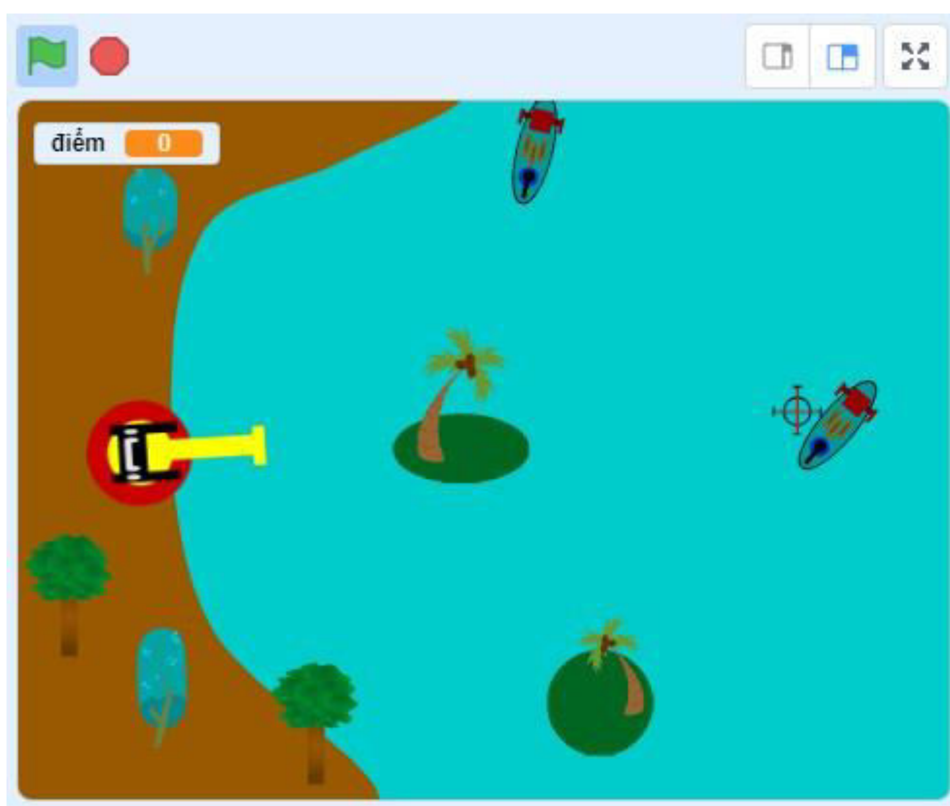
Chương trình đã được hoàn thiện, bây giờ chúng ta có thể lập trình một trò chơi cho riêng mình.

CHƯƠNG V

Bảo vệ đảo

Sản phẩm mà các bạn được hướng dẫn trong chương này là một trò chơi. Các bạn sẽ đóng vai các chiến sĩ, điều khiển khẩu pháo để bắn hạ các tàu địch đang muốn tiến tới xâm chiếm hai hòn đảo của chúng ta.

Tổng quan chương trình

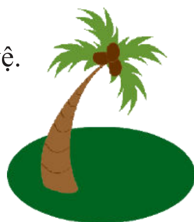


Trong trò chơi, các tàu địch sẽ xuất hiện ngẫu nhiên trên phần vùng biển của bản đồ và di chuyển đến hai đảo. Người chơi sẽ phải sử dụng chuột để bắn các tàu đó bằng cách nháy chuột vào tàu muốn bắn. Mỗi khi bắn trúng, tàu địch sẽ bị hạ và người chơi sẽ được cộng điểm. Nếu tàu địch chạm tới một trong hai đảo, trò chơi sẽ kết thúc.

Chương trình gồm có các đối tượng:

1. Đảo1

Đảo thứ nhất cần bảo vệ.



2. Đảo2

Đảo thứ hai cần bảo vệ.



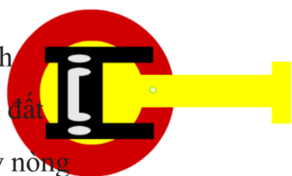
3.Pháo

Pháo nằm cố định

tại một vị trí trên đất

liền và luôn quay nòng

về phía con trỏ chuột.



4. Lửa

Lửa luôn được ẩ

và chỉ xuất hiện

trong một thời gian

ngắn khi người chơi nháy chuột để bắn.

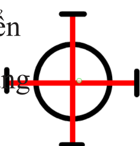


5.Tâm ngắm

Tâm ngắm luôn di chuyển

theo con trỏ chuột và trùng

với con trỏ chuột.



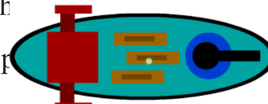
6.Tàuđịch

Tàuđịch

trí xuất p

ngẫu nhiên từ

biên bên trên hoặc biên bên phải của bản đồ và tiến về một trong hai đảo.



7.Chữ Start

Hiện lên khi nháy vào nút lệnh lá cờ xanh rồi

lại ẩn đi, báo hiệu trò chơi được bắt đầu.

8. Chữ Game Over

Được ẩn đi, đến khi Tàu địch chạm vào một trong hai đảo thì mới hiện lên, báo hiệu trò chơi kết thúc.

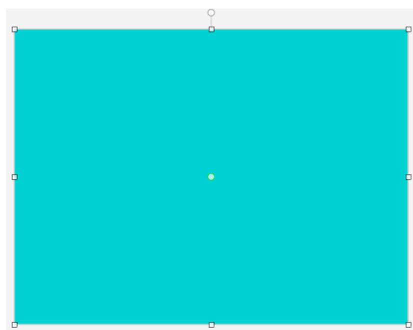
Không phải tất cả các chức năng của chương trình sẽ được hướng dẫn mà sẽ ómột số phần được đưa vào phần Bài tập thực hành ở cuối mỗi bài. Các bạn cần làm bàitập đầy đủ để hoàn thiện chương trình.

BÀI 1 VẼ BẢN ĐỒ

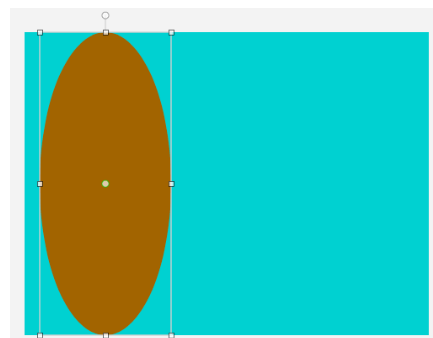
Trước tiên, chúng ta xoá **Chú mèo** có sẵn và vẽ bản đồ làm hình nền cho chương trình.

1.1 Vẽ biển và đất liền

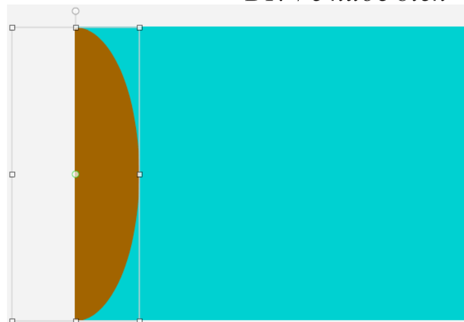
Chọn khung nền sân khấu vào Khu thiết kế phông nền, chuyển về chế độ Vector, vẽ một hình chữ nhật màu xanh nước biển kín trang vẽ. Phần đất liền được vẽ bằng hình elip màu nâu, chỉnh vị trí của hình ra lề trái và sử dụng công cụ **Chỉnh hình** như các bước ở *Hình 5.1.1.*



B1. Vẽ nước biển



B2. Vẽ hình elip



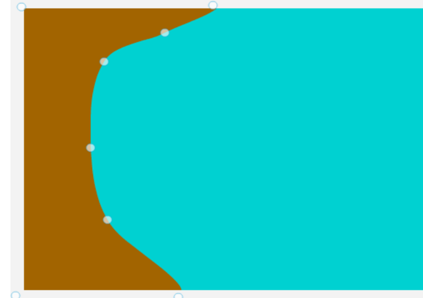
B3. Cắt hình elip



B4. Đưa hình elip sang lề trái

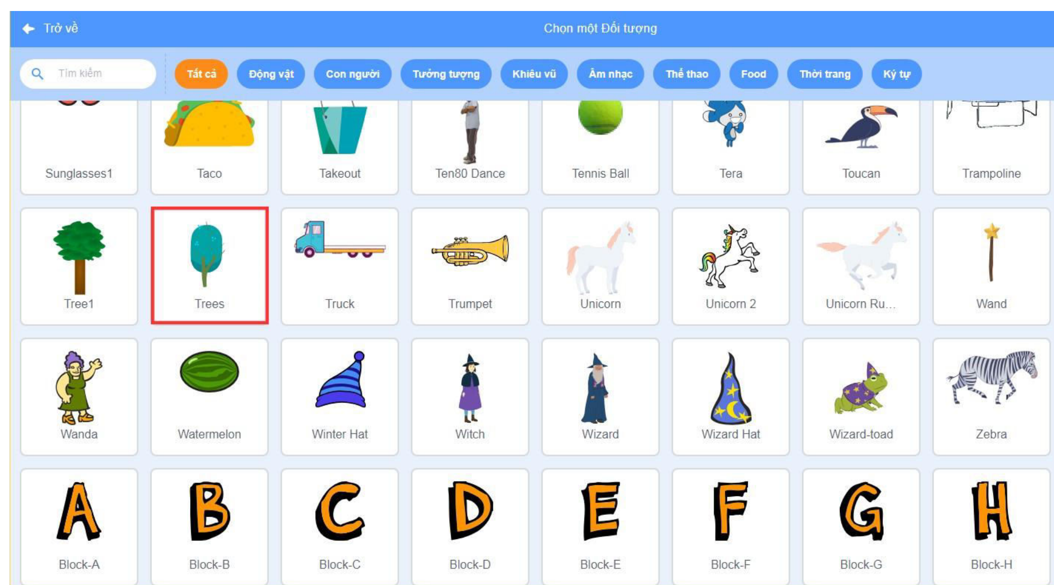


B5. Chỉnh lại đường biên

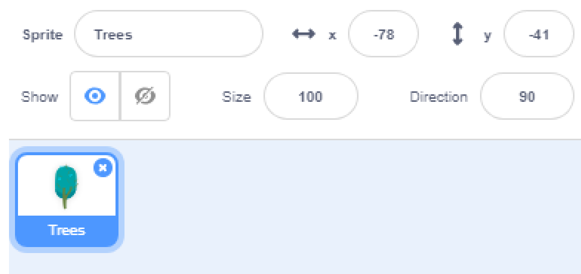


Hình 5.1.1. Vẽ biển và đất liền

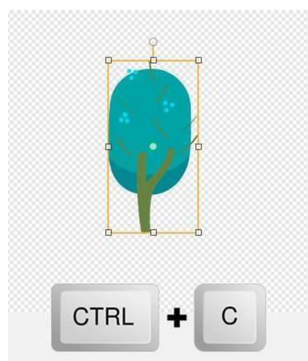
Thêm đối tượng cây từ thư viện ở Khung nhân vật.



Hình 5.1.2. Thêm nhân vật cây

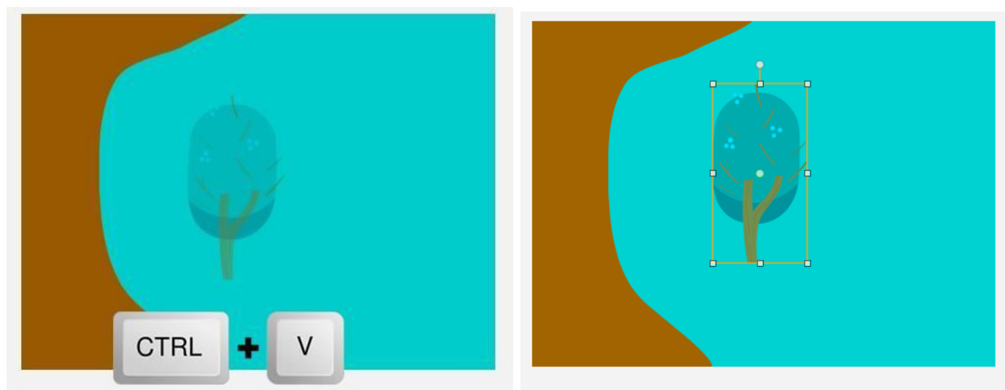


Để phông nền có hình ảnh cây, ta chuyển sang khu thiết kế của nhân vật cây , chọn cây và sao chép bằng công cụ hoặc nhấn tổ hợp phím Ctrl +C.



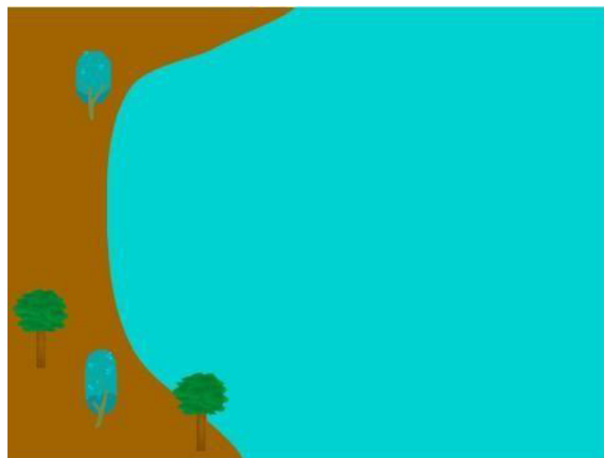
Hình 5.1.3. Sao chép hình ảnh cây

Chọn hình nền ở khu thiết kế phông nền, nhấn công cụ hoặc nhấn tổ hợp phím Ctrl + V, nháy chuột để dán hình cây và thay đổi kích thước cho hợp lí.



Hình 5.1.4. Dán hình và thay đổi kích thước

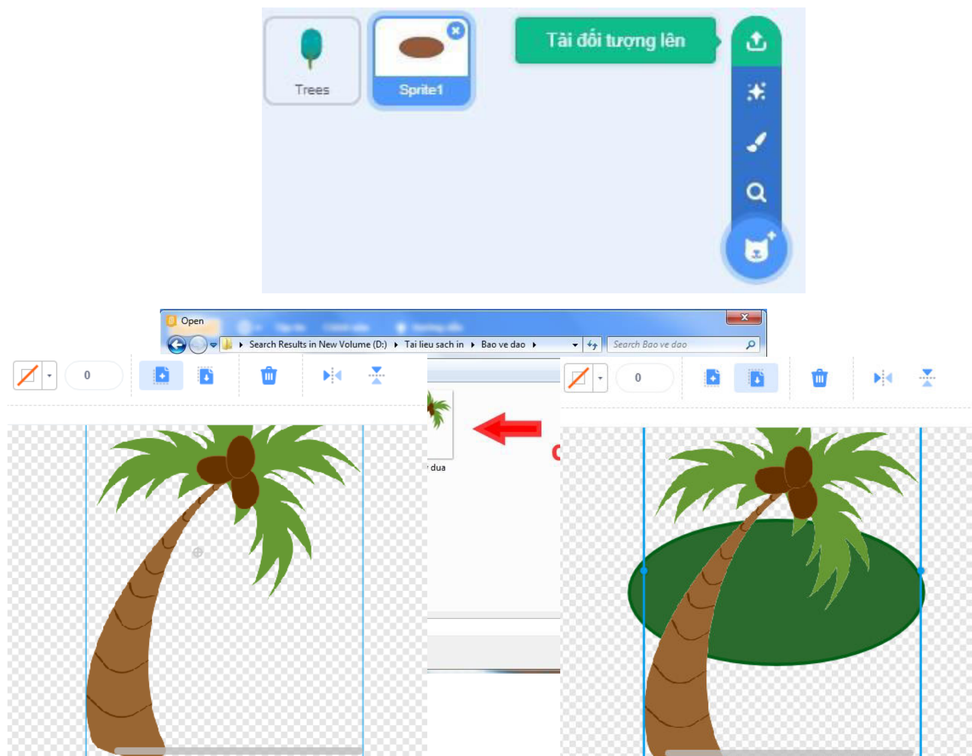
Ta có thể thêm nhiều cây khác bằng cách tương tự. Sau khi thêm xong ta chú ý xoá nhân vật Cây ở Khung nhân vật.



Hình 5.1.5. Đất liền và biển hoàn chỉnh

1.2. Vẽ hai đảo

Hai đảo là hai đối tượng riêng biệt, nên ta cần nháy vào biểu tượng ở khung nhân vật để vẽ thêm đối tượng. Đầu tiên vẽ **Đảo 1** bằng hình elip. Sau đó ta thêm vào cây dừa có sẵn trong tài liệu tải về bằng nút và sử dụng 2 công cụ và . Chỉnh kích thước cây dừa cho phù hợp.



Hình 5.1.6. Vẽ nền đảo và lấy hình ảnh cây dừa



Hình 5.1.7. Vẽ Đảo 1

Đảo 2 được vẽ tương tự, nhưng vẽ hình tròn thay cho hình elip và sử dụng công cụ để lật cây dừa theo hướng ngược lại.



Hình 5.1.9. Vẽ Đảo 2

BÀI 2 VẼ PHÁO VÀ TÀU ĐỊCH

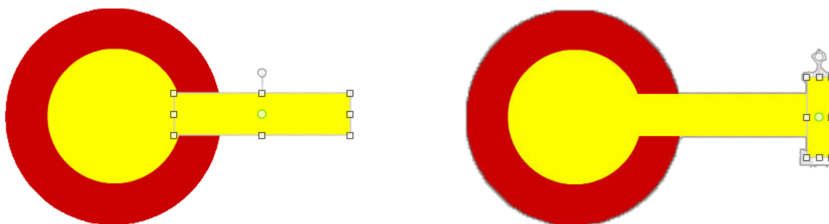
2.1.Vẽ Pháo

Ta thêm đối tượng mới bằng cách nháy vào biểu tượng và tiến hành vẽ pháo. Đầu tiên là vẽ thân pháo và tháp pháo bằng 2 hình tròn. Ta nhấn giữ phím **Shift** khi vẽ để vẽ được hình tròn.

Hình 5.2.1. Vẽ thân pháo

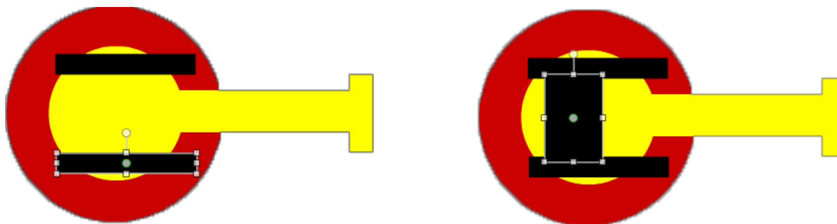


Tiếp theo là vẽ nòng pháo bằng hai hình chữ nhật.



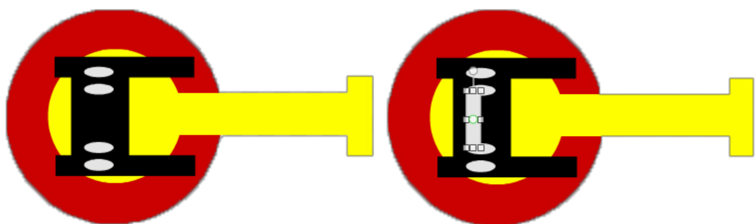
Hình 5.2.2. Vẽ nòng pháo

Hai súng bên trên tháp pháo sẽ được vẽ bằng ba hình chữ nhật. Chúng ta nên phóng to trang vẽ để vẽ dễ dàng và chính xác hơn.



Hình 5.2.3. Vẽ súng cho tháp pháo

Phần cuối cùng được vẽ bằng một hình chữ nhật và bốn hình elip. Sau khi vẽ xong một hình elip, các bạn sử dụng 2 công cụ **Sao chép** và **Dán** để tạo ra ba hình còn lại.



Hình 5.2.4. Hoàn thành vẽ pháo

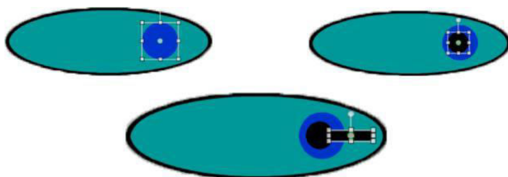
2.2. Vẽ Tàuđịch

Tàu địch là một đối tượng mới nên ta phải thêm nhân vật bằng cách nháy vào biểu tượng . Đầu tiên là vẽ thân tàu bằng một đường viền hình elip, sau đó đổ màu vàobên trong đường viền này.



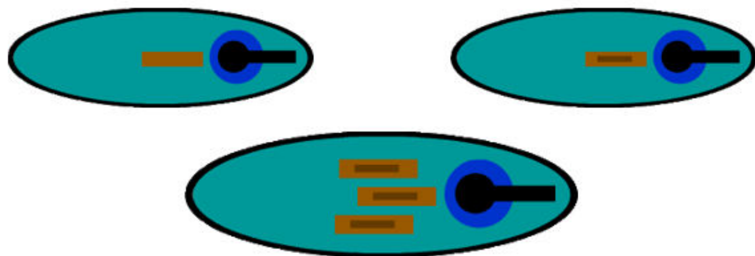
Hình 5.2.5. Vẽ thân Tàu địch

Vẽ pháo cho tàu, gồm hai hình tròn là thân pháo và tháp pháo cùng với một hình chữ nhật làm nòng pháo.



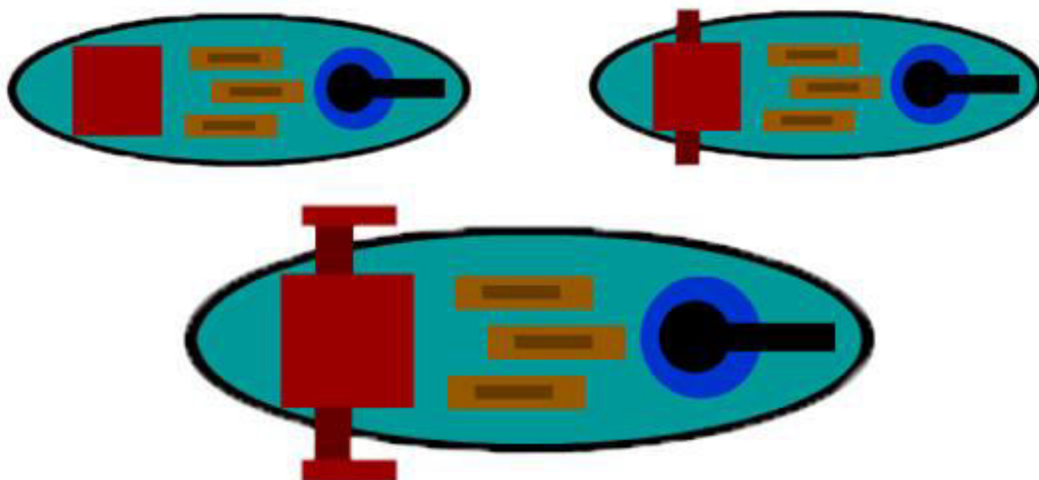
Hình 5.2.6. Vẽ pháo cho tàu

Phần giữa tàu gồm các hình chữ nhật, ta nên sử dụng công cụ **Sao chép** và **Dán** để vẽ nhanh hơn.



Hình 5.2.7. Trang trí phần thân tàu

Phần đuôi tàu được vẽ bằng một hình vuông và bốn hình chữ nhật. Tanên chỉnh sửa vị trí các hình bằng phím mũi tên trên bàn phím để sắp xếp chính xác hơn.



Hình 5.2.8: Hoàn thành vẽ Tàu địch

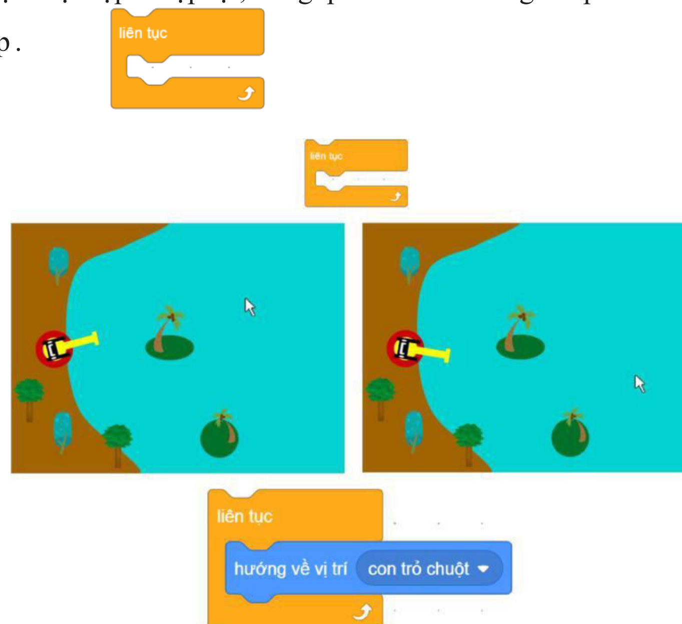
BÀI 3 LẬP TRÌNH ĐỐI TƯỢNG PHÁO VÀ TÂM NGẮM

Bài học này chúng ta sẽ lập trình cho đối tượng Pháo và Tâm ngắm. Lưu ý Tâm ngắm ta phải tự vẽ như trong phần bài tập thực hành của Bài 1.

3.1. Lập trình đối tượng Pháo

Trong chương trình, đối tượng Pháo luôn hướng nòng về con trỏ chuột. Sử dụng khối lệnh hướng về vị trí con trỏ chuột để điều khiển nòng pháo quay về phía con trỏ chuột.

Để khối lệnh được thực hiện lặp đi lặp lại, nòng pháo luôn hướng về phía con trỏ chuột, ta ghép thêm khối lệnh lặp .



Hình 5.3.1. Lập trình cho nòng pháo luôn xoay về phía con trỏ chuột

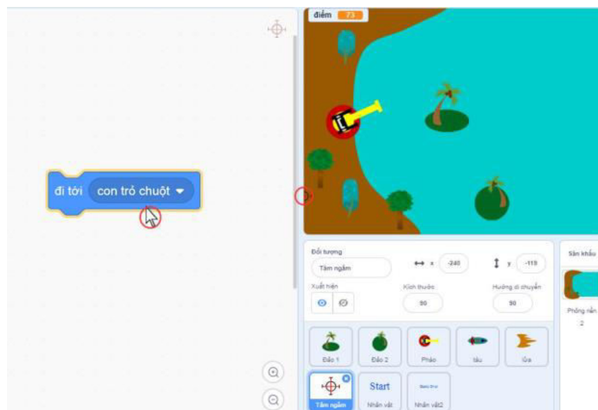
Đoạn khối lệnh hoàn chỉnh cho đối tượng Pháo như sau:



Hình 5.3.2. Đoạn khối lệnh hoàn chỉnh

3.2. Lập trình đối tượng Tâm ngắm

Tâm ngắm luôn di chuyển theo con trỏ chuột. Ta sử dụng khối lệnh để tâm ngắm di chuyển theo con trỏ chuột. 

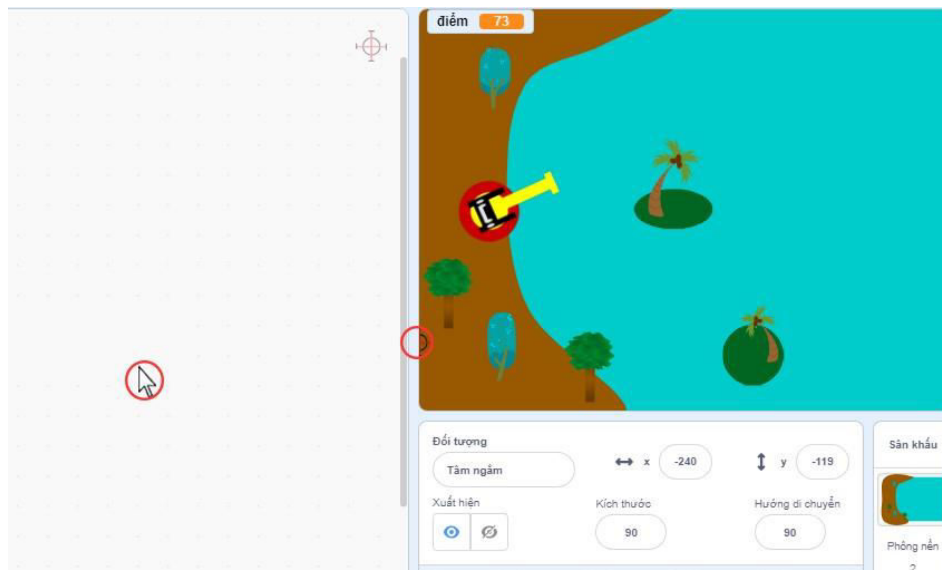


Hình 5.3.3. Chạy thử khối lệnh Nhảy tới con trỏ chuột

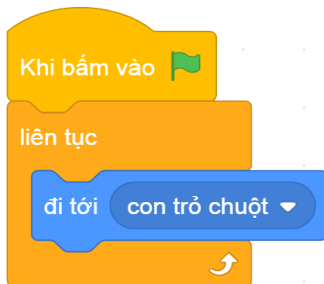
Khi chạy thử khối lệnh, con trỏ chuột đang ở ngoài sân khấu, trong khi các đối tượng chỉ được phép di chuyển bên trong sân khấu. Do đó, Tâm ngắm chỉ di chuyển đến biên rồi dừng lại. Tương tự như đối tượng Pháo ta cho vào khối lệnh lặp  để Tâm ngắm luôn di chuyển theo con trỏ chuột.



Khi con trỏ chuột trong sân khấu



Khi con trỏ chuột ra ngoài sân khấu



Hình 5.3.4. Kết quả và đoạn khối lệnh hoàn chỉnh

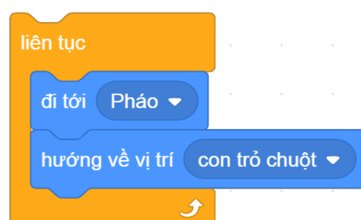
BÀI 4 LẬP TRÌNH CHO LỬA

Cũng như vẽ Tâm ngắm, vẽ Lửa sẽ không được hướng dẫn mà được đưa vào bài tập thực hành ở Bài 2. Bài này chúng ta sẽ lập trình cho đối tượng Lửa.

4.1. Lập trình vị trí của Lửa

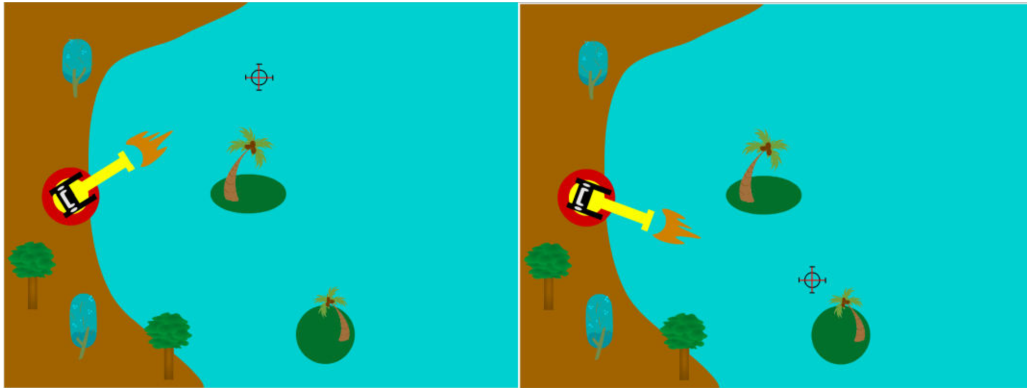
Đầu tiên, lập trình cho Lửa di chuyển tới Pháo, luôn hướng về phía con trỏ chuột. Đây là lời giải của phần Bài tập thực hành ở Bài 3.





Hình 5.4.1. Lập trình cho Lửa nhảy tới Pháo và hướng về con trỏ chuột

Tiếp theo, lập trình điều khiển Lửa luôn xuất hiện ở đầu nòng pháo chứ không phải ở giữa thân pháo.



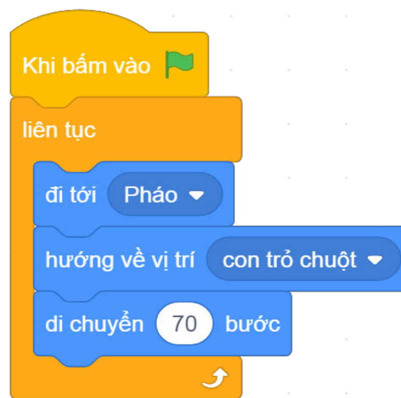
Hình 5.4.2. Vị trí của Lửa so với Pháo

Cả hai đối tượng Lửa và Pháo đều cùng quay về phía con trỏ chuột. Lập trình vị trí của Lửa không nằm ở thân Pháo mà ở gần đầu nòng Pháo bằng khối lệnh , số bước di chuyển ( ) bước cần di chuyển tuỳ thuộc vào độ dài của nòng pháo.



Hình 5.4.3. Di chuyển Lửa

Vì hướng của nòng pháo luôn hướng về con trỏ chuột và thay đổi liên tục, vậy nên đoạn khối lệnh điều khiển việc nhảy tới Pháo, hướng về phía con trỏ chuột và di chuyển của Lửa phải được thực hiện liên tục bằng khối lệnh lặp.



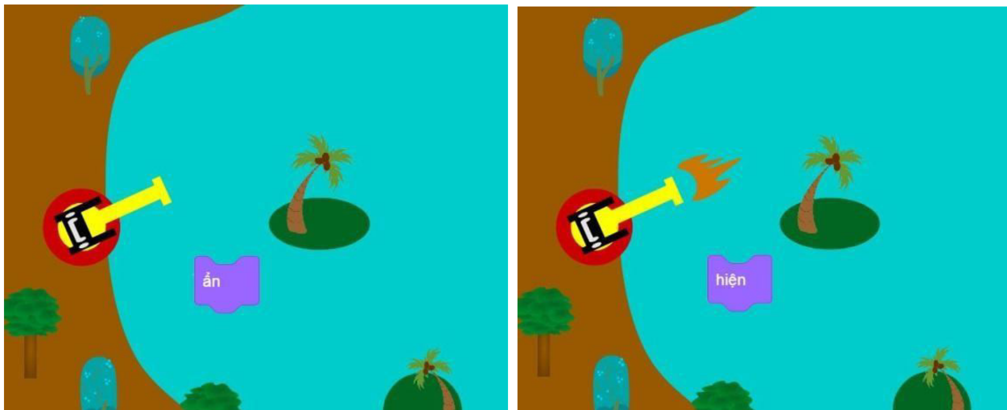
Hình 5.4.4. Đoạn khối lệnh điều khiển vị trí của Lửa

Khi chạy chương trình, Lửa luôn di chuyển ra phía nòng pháo, nếu di chuyển chuột, nòng pháo và Lửa luôn hướng về phía con trỏ chuột.

4.2. Lập trình đối tượng Lửa chỉ hiện khi nháy chuột

Trong chương trình, khi chúng ta nháy chuột để bắn, Lửa sẽ xuất hiện trong một khoảng thời gian rất ngắn rồi lại biến mất. Để Lửa biến mất, chúng ta sử dụng khối lệnh

ẩn , khi muốn hiện trở lại, sử dụng khối lệnh hiện.



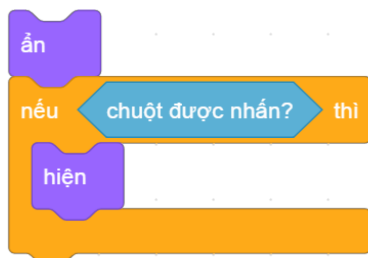
Hình 5.4.5. Khối lệnh ẩn và hiện

Khối lệnh chuột được nhấn? trong nhóm **Cảm biến** sẽ giúp chúng ta kiểm tra xem chuột đã nháy hay chưa. Khối lệnh này trả về cho chúng ta giá trị **false** (sai) nếu chưa nháy chuột, giá trị **true** (đúng) nếu nháy chuột.

Chúng ta lập trình cho đối tượng Lửa bằng cách sử dụng khối lệnh **Nếu – thì**

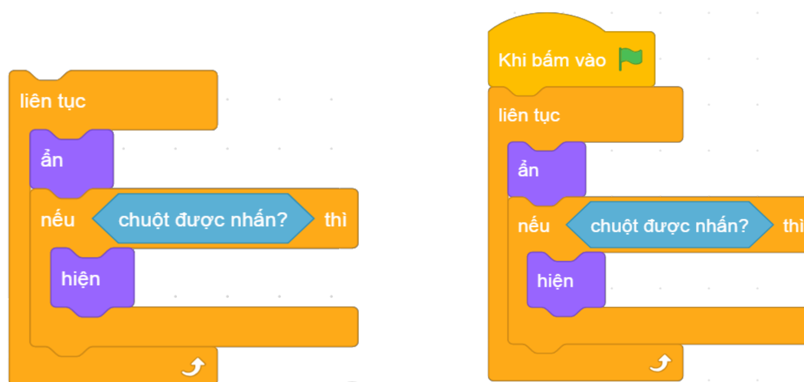


xác định thời điểm ẩn, hiện của đối tượng. Ở trạng thái ban đầu, Lửa được ẩn đi, nếu nháy chuột thì hiện Lửa lên.



Hình 5.4.6. Đoạn khối lệnh kiểm tra

Ghép thêm khối lệnh **Lặp** để công việc kiểm tra xem có nháy chuột hay không được thực hiện trong suốt chương trình.



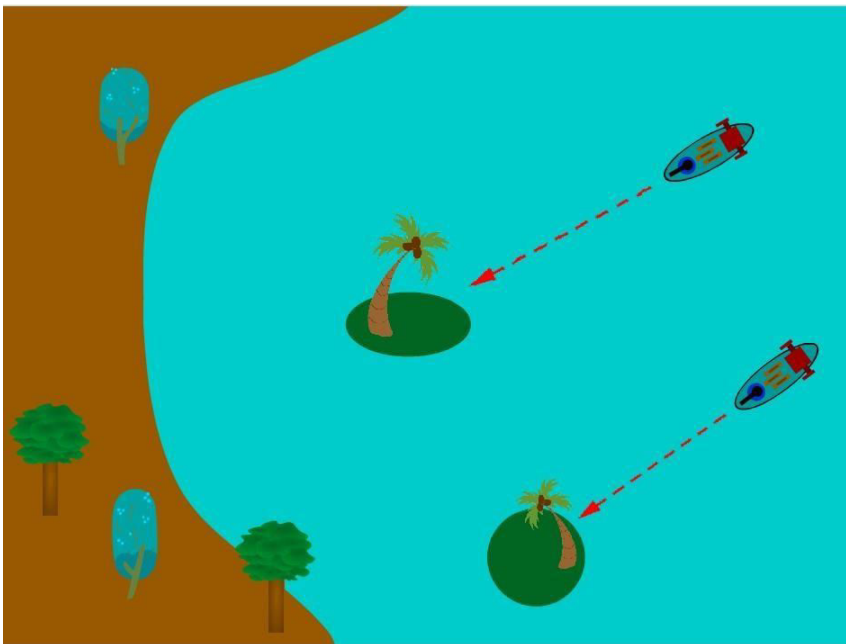
Hình 5.4.7. Đoạn khối lệnh điều khiển sự ẩn, hiện của Lửa

BÀI 5 LẬP TRÌNH CHO TÀU ĐỊCH

Lập trình đối tượng Tàu địch tương đối phức tạp nên được chia ra làm nhiều bài học để các bạn dễ dàng nắm bắt hơn.

5.1. Vị trí của tàu khi hướng về Đảo 1

Tàu địch xuất hiện ở vị trí ngẫu nhiên từ phía bên phải hoặc phía trên của sân khấu. Trong chương trình, tàu xuất hiện ở phía trên sân khấu sẽ tiến đến Đảo 1, còn Tàu xuất hiện ở bên phải sân khấu sẽ tiến đến Đảo 2.



Hình 5.5.1. Vị trí và hướng của Tàu địch

Ta lập trình cho đối tượng Tàu địch. Việc Tàu xuất hiện ở phía trên hay bên phải làngẫu nhiên. Tạo biến tên là `hướng`, đặt cho biến giá trị ngẫu nhiên 1 hoặc 2.

`đặt hướng thành lấy ngẫu nhiên từ 1 đến 2`. Kiểm tra giá trị của biến ***hướng***, `hướng = 1` nếu ***hướng*** bằng 1 thì tàu sẽ xuất hiện ở bên trên và hướng về Đảo 1, còn nếu không thì Tàu sẽ xuất hiện ở bên phải và hướng về Đảo 2.

Sử dụng khối lệnh `nếu … thì / nếu không thì` để kiểm tra tàu địch sẽ đi về hướng Đảo 1 hay Đảo 2.

Khối lệnh được mô tả như trong *Hình 5.5.2.*

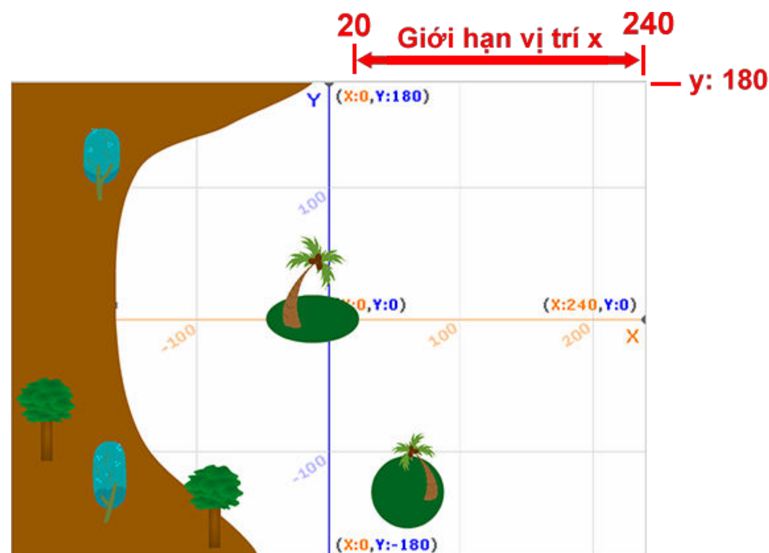


Hình 5.5.2. Khối lệnh Nếu – thì – nếu không thì

Khối lệnh sẽ kiểm tra nếu biến ***hướng*** bằng 1 thì sẽ thực hiện công việc 1 Làm công việc 1, còn nếu không (***hướng*** khác 1) thì sẽ thực hiện công việc 2 Làm công việc 2. Trong trường hợp này, nếu ***hướng*** bằng 1 thì Tàu địch hướng về vị trí Đảo 1 hướng về vị trí Đảo 1 và xuất hiện tại vị trí bên trên.

Khối lệnh hướng về vị trí sẽ điều khiển Tàu xoay về phía nhân vật khác, trong trường hợp này là Đảo 1. Vị trí biến trên có $y = 180$ và x thay đổi được từ 20 đến 240, lấy ngẫu nhiên giá trị trong khoảng 20 đến 240 và ghép vào giá trị x trong khối lệnh đi tới

đi tới điểm x: lấy ngẫu nhiên từ 20 đến 240 y: 180



Hình 5.5.3. Tàu địch xuất hiện ở vị trí biên phía trên sân khấu

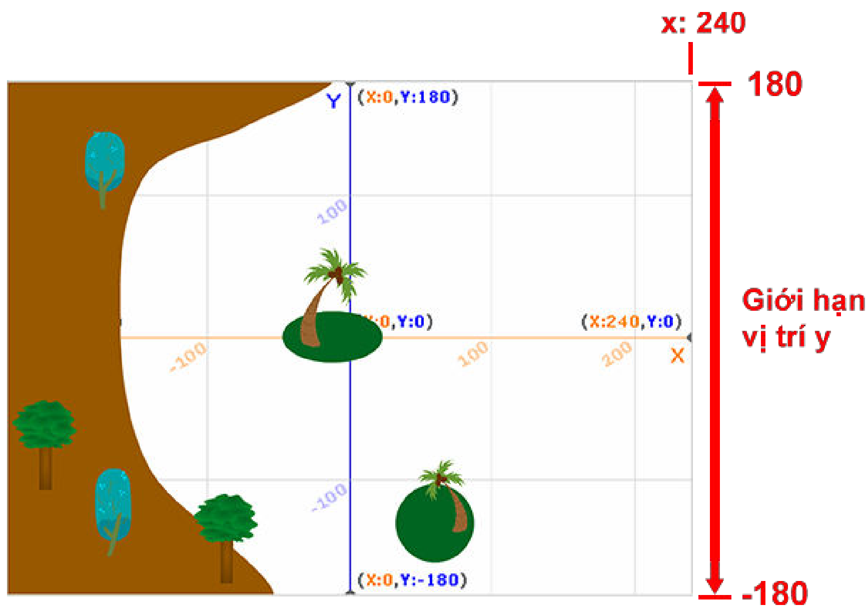


Hình 5.5.4. Lập trình cho Tàu địch trong trường hợp là tàu tiến về Đảo 1

5.2. Vị trí của tàu khi hướng về Đảo 2

Tương tự như vậy, nếu *hướng* khác 1 thì Tàu địch nhảy tới vị trí bất kì ở biên bên phải và xoay về phía Đảo 2 [hướng về vị trí Đảo 2]. Biên bên phải có vị trí *x* = 240 và có *y* thay đổi từ 180 đến – 180, vậy ta cần khối lệnh





Hình 5.5.5. Tàu địch xuất hiện ở vị trí biên phải sân khấu



Hình 5.5.6. Đoạn khối lệnh hoàn chỉnh cho sự xuất hiện của Tàu địch

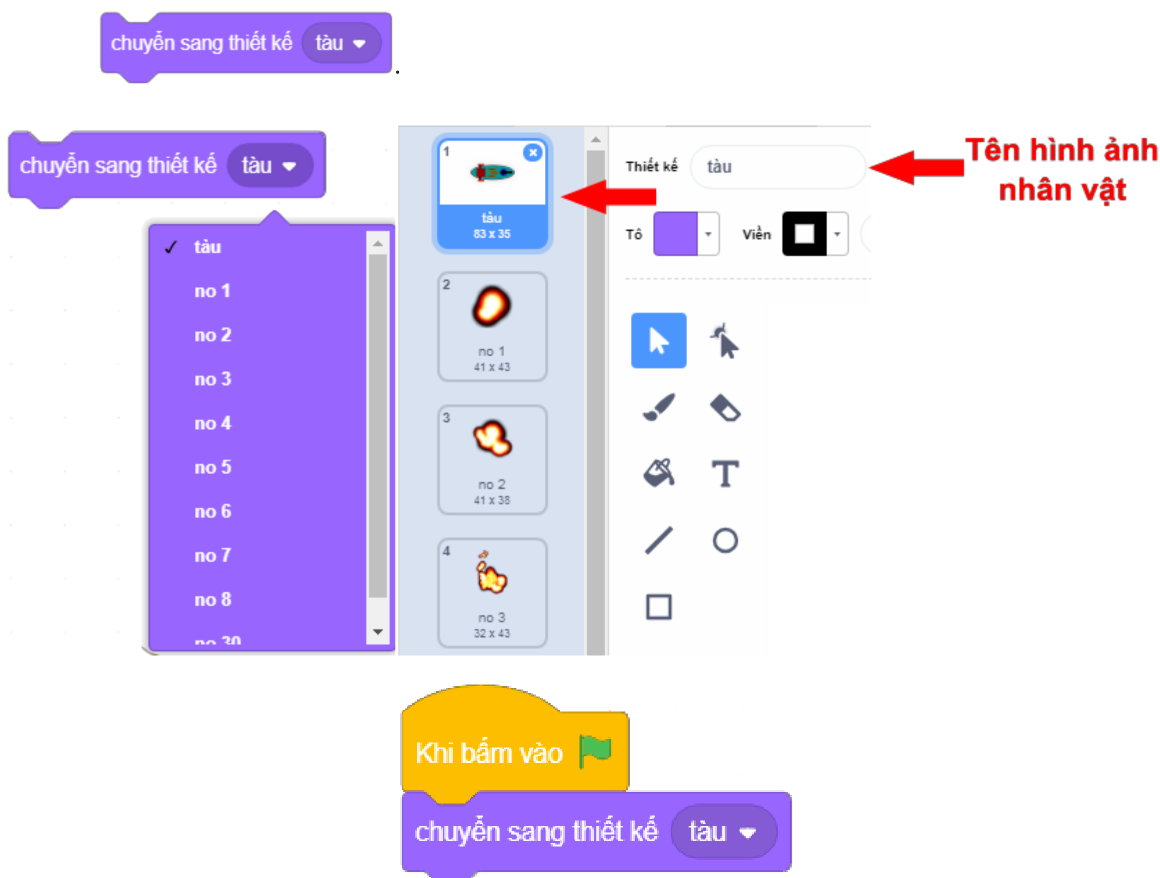
BÀI 6 LẬP TRÌNH CHO TÀU ĐỊCH (TIẾP THEO)

Trong chương trình sẽ xuất hiện rất nhiều tàu địch. Trong bài học này các bạn sẽ được hướng dẫn cách tạo bản sao cho Tàu địch và điều khiển những bản sao đó d chuyển tới mục tiêu.

Lập trình cho bản sao

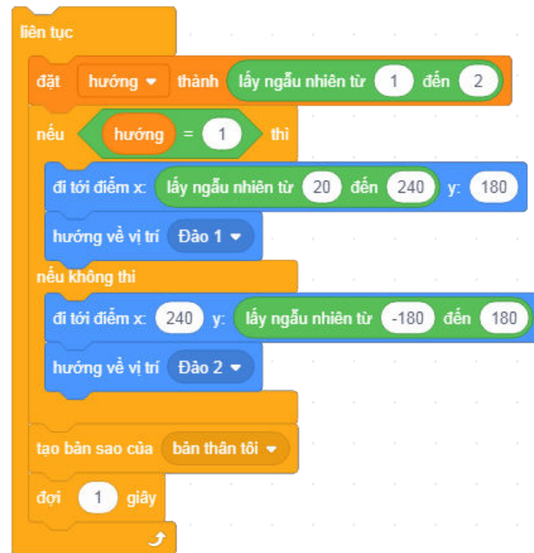
Ở bài tập thực hành của bài 5, ta đã thêm hình ảnh để mô phỏng hiện tượng nổ cho nhân vật Tàu địch. Những hình ảnh này sẽ được sử dụng khi tàu bị bắn và sẽ được hướng dẫn trong bài sau.

Khi bắt đầu chương trình, đổi lại hình ảnh nhân vật Tàu địch về trạng thái ban đầu bằng khối lệnh



Hình 5.6.1. Đặt hình ảnh nhân vật về hình ảnh ban đầu

Sau khi di chuyển đến vị trí thích hợp và hướng về phía một trong hai quần đảo, Tàu địch sẽ tạo ra bản sao của chính nó bằng khối lệnh tạo bản sao của bản thân tôi .Bản chính sẽ phải lặp lại việc nhảy đến vị trí thích hợp và tạo ra bản sao một cách liên tục, mỗi lần cách nhau 1 giây.



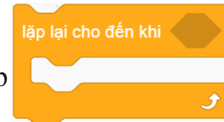
Hình 5.6.2. Tạo bản sao của đối tượng Tàu địch

Lúc này thì bản sao của tàu đã được tạo ra nhưng chưa thực hiện công việc gì cả. Để lập trình cho bản sao di chuyển, ta cần sử dụng khối lệnh khi tôi bắt đầu là một bản sao

Với khối lệnh này, khi một bản sao được tạo ra, tất cả các khối lệnh ghép dưới khối lệnh này sẽ được thực hiện cho bản sao đó.

Khi bản sao được tạo, hướng của bản sao cũng chính là hướng của bản chính, trong khi bản chính đã hướng sẵn về phía một trong hai quần đảo, vậy bản sao chỉ cần di chuyển về phía trước bằng khối lệnh di chuyển 2 bước

Bản sao sẽ di chuyển cho đến khi chạm vào Đảo 1 hoặc Đảo 2 thì dừng.

Trong trường hợp này chúng ta sẽ sử dụng khối lệnh lặp 

.

Ta hãy xem xét cách sử dụng khối lệnh lặp này trong *Hình 5.6.3.*

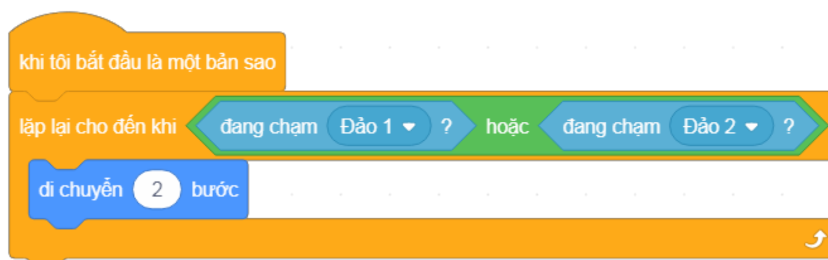


Hình 5.6.3. Khối lệnh Lặp cho tới khi

Khi chạy đến khối lệnh này, khối lệnh Điều kiện sẽ được kiểm tra xem là **đúng** hay sai. Nếu sai thì khối lệnh Làm công việc 1 được thực hiện lặp đi lặp lại, cho đến khi trở thành đúng thì sẽ dừng vòng lặp và thực hiện Điều kiện

Bản sao của Tàu địch sẽ sử dụng khối lệnh như sau:

Hình 5.6.4. Đoạn khối lệnh điều khiển bản sao di chuyển



Trong trường hợp này, tàu sẽ lặp lại việc di chuyển cho đến khi

đang chạm Đảo 1 ? hoặc đang chạm Đảo 2 ? thì dừng lại. Ở đây, khối lệnh đang chạm Đảo 1 ? sẽ kiểm tra xem đối tượng Tàu địch này có chạm vào đối tượng Đảo 1 hay không, nếu có sẽ trả về **true** (đúng), còn nếu không thì sẽ là giá trị **false** (sai).

Tương tự, khối lệnh đang chạm Đảo 2 ? sẽ kiểm tra xem Tàu có chạm vào đối tượng Đảo 2 hay không.

Chỉ cần một trong hai khối lệnh kiểm tra có chạm vào đảo hay không trả về giá trị **true** (đúng) thì cả khối lệnh đang chạm Đảo 1 ? hoặc đang chạm Đảo 2 ? sẽ mang giá trị đúng, tức là việc di chuyển sẽ được thực hiện liên tục cho đến khi tàu chạm vào một trong hai đối tượng Đảo 1 hoặc Đảo 2 (hoặc Đảo 1 hoặc Đảo 2) thì dừng lại.

BÀI 7 LẬP TRÌNH CHO TÀU ĐỊCH (TIẾP THEO)

Trong bài học này, các bạn sẽ lập trình để Tàu địch bị bắn trúng sẽ nổ và người chơi được cộng điểm.

Lập trình cho Tàu địch bị bắn

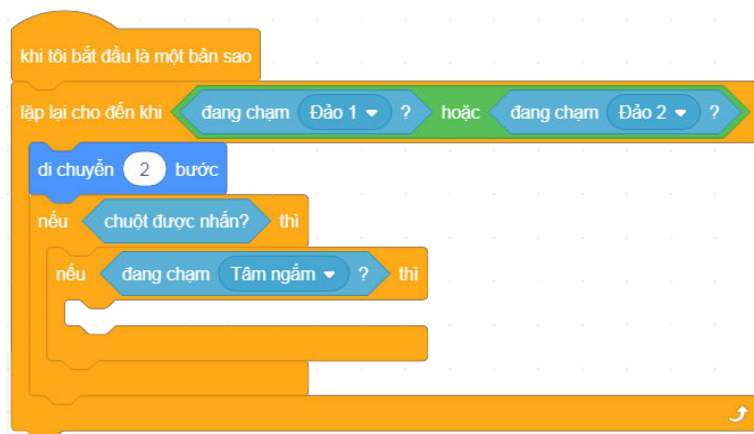
Người chơi sẽ bắn Tàu địch bằng cách nháy chuột lên tàu. Trong khi bản sao của tàu di chuyển đến Đảo 1 hoặc Đảo 2, nếu Pháo bắn (người chơi nháy chuột), mà tâm ngắm lại chạm bản sao (tàu bị bắn trúng) thì điểm sẽ được cộng, bản sao tàu sẽ nổ và biến mất.

Bản sao của Tàu địch sẽ kiểm tra việc Pháo bắn hay không trong lúc di chuyển về phía quần đảo.

Hình 5.7.1. Kiểm tra xem Pháo có bắn hay không



Khối lệnh chuột được nhấn? sẽ trả về giá trị **true** (đúng) nếu người chơi nháy chuột. Khi người chơi nháy chuột, (khối lệnh chuột được nhấn? có giá trị đúng) thì kiểm tra tiếp xem Tàu địch có đang chạm vào Tâm ngắm (Pháo bắn trúng) không.



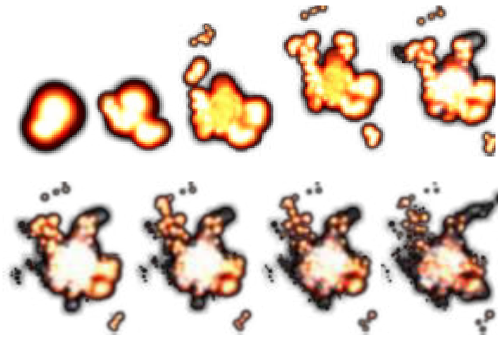
Hình 5.7.2. Kiểm tra xem Pháo có bắn trúng Tàu địch không

Nếu Pháo bắn trúng (người chơi nháy chuột) vào Tàu địch thì điểm sẽ được cộng thêm 1 và

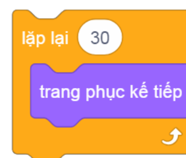
điểm

tàu sẽ nổ rồi biến mất. Trong phần bài tập thực hành của bài 6, ta đã tạo biến , để tăng biến **điểm** lên ta sử dụng khối lệnh 

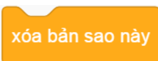
Từ bài học thực hành của bài 5, ta đã thêm 30 hình ảnh vào **Khu ngoại hình** để thể hiện hình ảnh từ khi phát nổ đến khi tro tàn của Tàu địch, ta chỉ cần lập trình để thay đổi lần lượt qua 30 hình ảnh này liên tục.



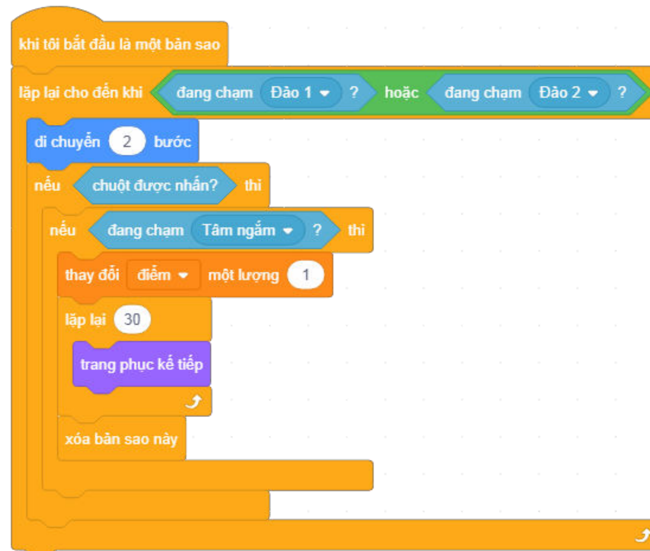
Hình 5.7.3. Một số hình ảnh nhân vật tàu phát nổ



Hình 5.7.4. Lập trình cho tàu phát nổ

Sau khi nổ xong vẫn còn lại hình tro tàn. Chúng ta phải xoá bản sao này đi bằng khối lệnh 

. Cuối cùng ghép các khối lệnh này lại với nhau.



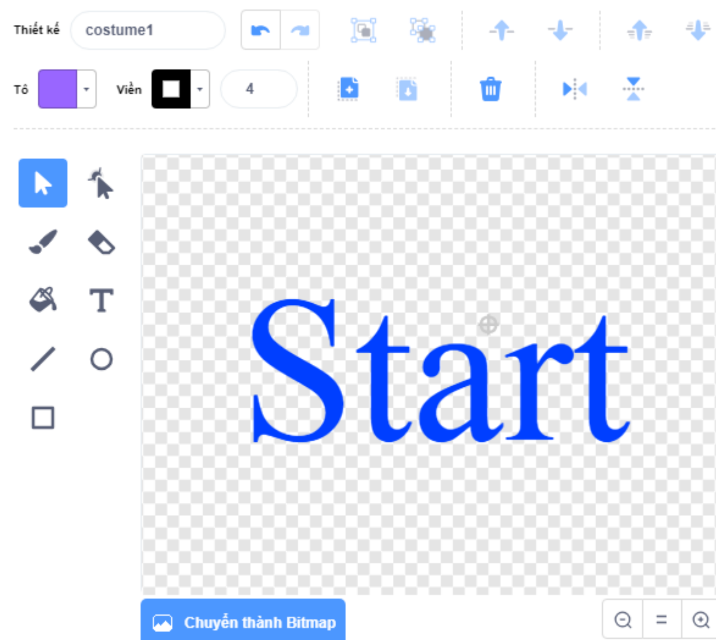
Hình 5.7.5. Thêm đoạn khối lệnh cho bản sao của Tàu địch

BÀI 8 KẾT THÚC CHƯƠNG TRÌNH

Đến đây chương trình đã gần được hoàn thành. Bài cuối cùng này sẽ hoàn thiện thêm để chương trình có bắt đầu và kết thúc rõ ràng.

8.1. Lập trình đối tượng chữ Start

Khi nháy vào nút lệnh lá cờ xanh bắt đầu chương trình, chữ Start sẽ hiện lên giữa sân khấu, báo hiệu trò chơi được bắt đầu. Ta cần thiết kế đối tượng này bằng cách nháy vào biểu tượng ở **Khung nhân vật** rồi chuyển đến **Khu ngoại hình**. Chúng ta sẽ sử dụng công cụ để thêm chữ Start. Ta có thể chỉnh kích thước, màu sắc, phông chữ tuỳ ý.



Hình 5.8.1. Thêm đối tượng chữ Start

Khi bắt đầu, đối tượng chữ Start này sẽ hiện lên trong vòng 1 giây rồi biến mất.

Chúng ta sẽ sử dụng lần lượt các khối lệnh

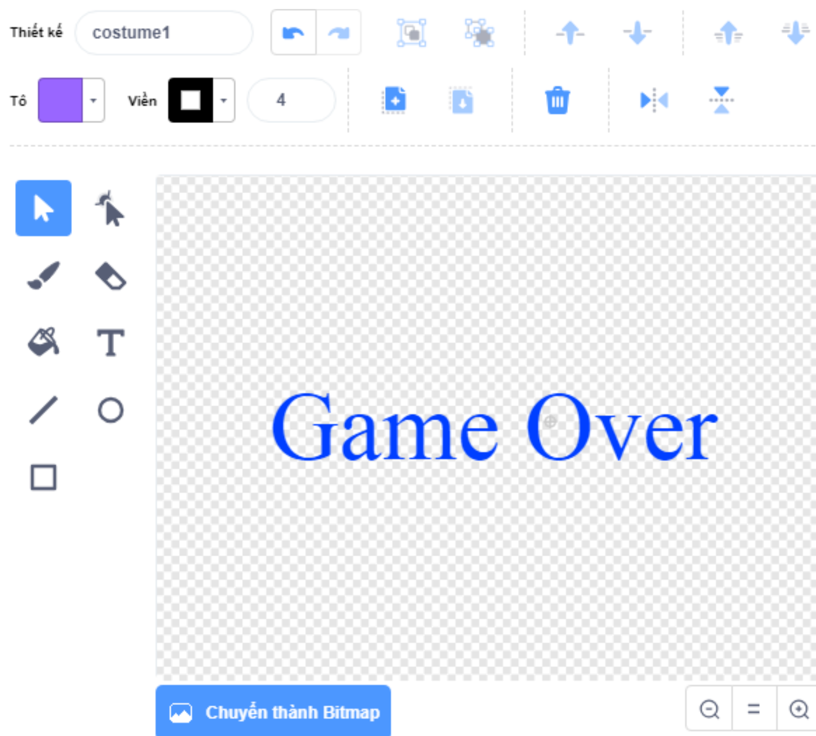




Hình 5.8.2. Lập trình cho đối tượng Start

8.2. Lập trình đối tượng chữ Game Over

Việc thêm mới và thiết kế đối tượng chữ Game Over cũng làm tương tự như đối tượng chữ Start.



Hình 5.8.3. Thêm đối tượng chữ Game Over.

Tính từ lúc bắt đầu chạy, trong suốt chương trình đối tượng chữ Game Over luôn được ẩn đi.



Hình 5.8.4. Đoạn khối lệnh ẩn đối tượng Game Over

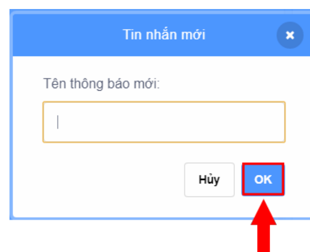
Đến khi kết thúc chương trình chữ Game Over sẽ hiện lên. Để đối tượng chữ Game Over này biết được thời điểm kết thúc chúng ta cần lập trình cho bản sao của tàu thông báo kết thúc tới tất cả các đối tượng khác trong chương trình khi chạm vào một trong hai đảo.

Để làm thông báo, ta sẽ sử dụng khối lệnh phát tin message1 và tạo một thông báo mới bằng cách nháy vào mũi tên trong khối lệnh và chọn *Thông báo mới*.



Hình 5.8.5. Tạo thông báo

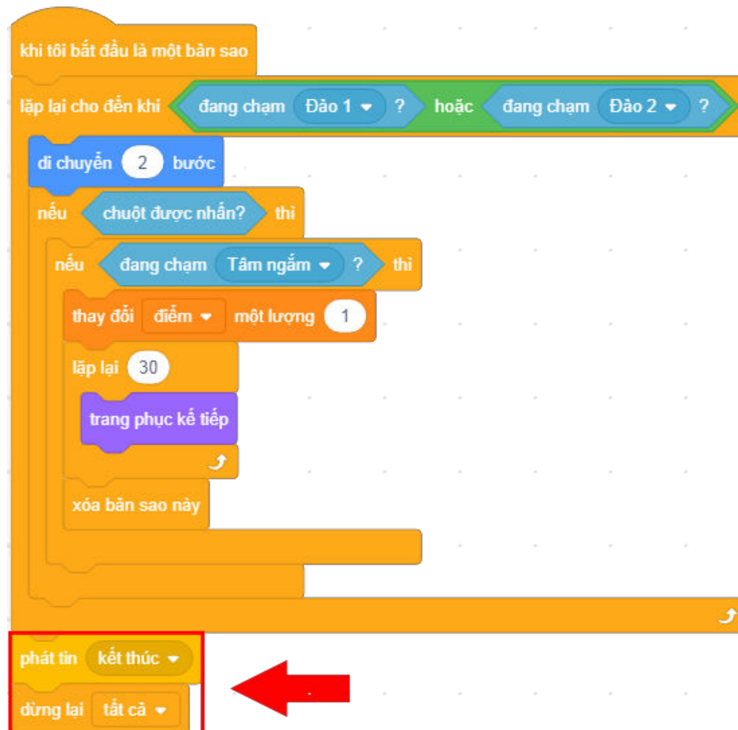
Lúc này, một cửa sổ sẽ hiện ra để bạn đặt tên cho thông báo. Sau khi đặt tên xong các bạn nhấn nút OK



Hình 5.8.6. Đặt tên cho thông báo

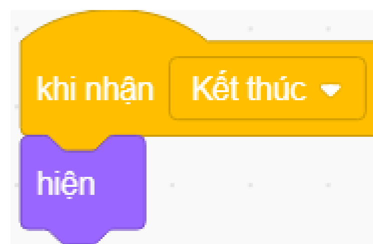
Các bạn ghép khối lệnh phát tin kết thúc xuống dưới khối lệnh lập trình cho bản sao của đối tượng Tàu địch.

Ở vị trí này, khi điều kiện đang chạm Đảo 1 ? hoặc đang chạm Đảo 2 ? đúng thì khối lệnh thông báo này sẽ được chạy. Sau khi thông báo xong chương trình sẽ ngừng bằng khối lệnh dừng lại tất cả . Khối lệnh này cũng là đáp án câu 1 phần bài tập thực hành của bài 7.



Hình 5.8.7. Đoạn khối lệnh hoàn chỉnh của bản sao đối tượng Tàu địch

Bây giờ tất cả các đối tượng khác đều có thể nhận được thông báo kết thúc bằng khối lệnh khi nhận kết thúc. Ta sẽ phải lập trình cho đối tượng chữ Game Over để khi nhận được thông báo kết thúc này sẽ hiện lên trên sân khấu.



Hình 5.8.8. Đoạn khối lệnh nhận thông báo

Vậy là chúng ta đã lập trình xong cho chương trình “Bảo vệ đảo”. Các bạn cần hoàn thành đầy đủ bài tập ở cuối mỗi bài để chương trình hoàn thiện hơn. Các bạn cũng có thể tự sáng tạo thêm các loại tàu khác, cũng như tăng số lượng và tốc độ của tàu để trò chơi thú vị hơn.

Bài tập nâng cao

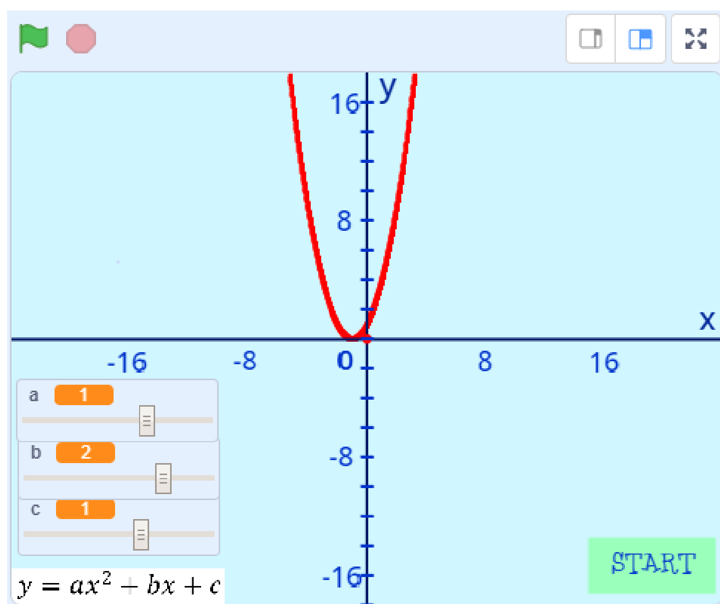
Sau khi học và hoàn thành các ví dụ trong phần nội dung cuốn sách, các bạn tiếp tục thử sức với các chương trình như sau:

(Mã nguồn của một số chương trình các bạn có thể tìm thấy trên trang web https://scratch.mit.edu/)

1. Trò chơi nổi tiếng của Nguyễn Hà Đông - Flappy Bird



2. Chương trình Vẽ đồ thị



3. Chương trình Máy tính Casio (Calculator)



4. Chương trình Giải phương trình bậc 2

